Università
Ca'Foscari
Venezia

Master's Degree

in Computer Science

Final Thesis

Querying and Clustering on Knowledge Graphs

A Dominant-Set based approach

Supervisor
Prof. Sebastiano Vascon

Co-supervisor
Prof. Marcello Pelillo

Graduand
Christian Bernabe Cabrera
843382

Academic Year
2019 / 2020

Contents

Acknowledgements

I would like to acknowledge everyone who played a role in my academic accomplishments.

My mother and my brother, who supported me with love and understanding.

My fiancee Greta, without your help, patience and advice I could have never reached this level of success.

My friends, each of whom accepted my long five minutes calls.

Finally my supervisors, that provided constant support and guidance throughout the project.

Gracias a quien estuvo y a quien está.

Chapter 1

Introduction

1.1 MEMories and EXperiences for inclusive digital storytelling

MEMEX H2020 (research and innovation programme under grant agreement No 870743) is a three years European project that will develop artificial intelligence methods for helping the fragile people in our society to be socially included again in Europe.

Figure 1.1: Logo MEMEX Project (image from [2])

It will be developed together with the communities at risk of social exclusion living in Lisbon, Barcelona and Paris. MEMEX's aim is to create an app that will show the stories of these people, linked to the place where they live and the cultural heritage that surrounds them, using augmented reality. In these way MEMEX project will be able to promote social cohesion through the access to Cultural Heritage (CH) related tools [2].

1.1.1 Social goal

MEMEX social goal is to encourage social cohesion to promote recognition of differences through the implementation of best practices for social inclusion and audience engagement strategies.This objectives will be achieved by throughout:

- **Social analysis for communities need**: The objective will suggest best practises to understand and study the causes of exclusion.
- **Guidelines to benchmark social inclusion:** Starting from the communities needs the task is to research and apply best practices for promoting

inclusion and cohesion with the usage of heritage-related tools given the cultural background of the community.

- **Design audience development strategies:** throughout the usage of the storytelling tools, the objective of MEMEX is to increase the interest of people for cultural events and diversify the audience.

1.1.2 Technologies

MEMEX promotes the usage and creation of new easy to use Information and communication technology (ICT) tools with the usage of Artificial Intelligence [2]. The main goals are:

- **Knowledge Graph infrastructure:** The objective is to create a novel KG and retrieval tools to be used in the geolocalised storytelling engine of MEMEX.
- **Geolocalization of cultural heritage:** The focus will be to develop a software running on mobile devices with the users knowledge and memories.
- **Storytelling with Augmented Reality:** Through the usage of AR, the goal is to create a compelling experience of the stories.

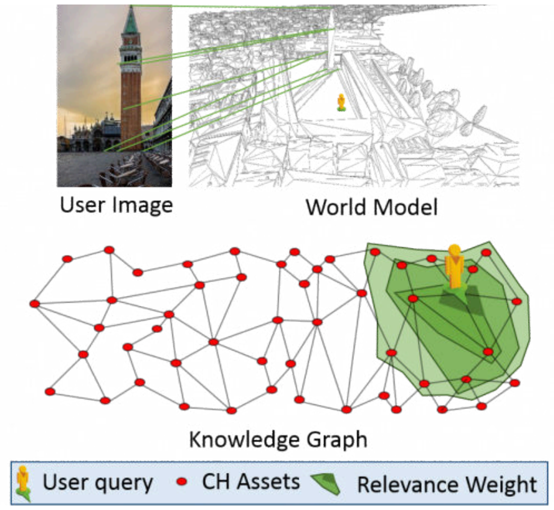


Figure 1.2: Example of the main goals (image from [2])

The technological embodiment of MEMEX is an app installed on a smartphone allowing non-expert users to create and visualise stories related to their personal memories and experiences digitally linked to the geographical locations of either intangible (e.g. an event) or a tangible cultural places/object. The user will be able, using Augmented Reality (AR), to annotate any physical object or location, with its memories, knowledge or stories about it through textual input, digital images or videos. Then, the targeted communities, that are thought to be made up of people who are systematically blocked by various cultural opportunities and resources, will be able to connect their experiences and memories with a new Knowledge Graph (KG), linking CH items and places with stories that are bound and entangled within the European history [2].

1.2 Stage

1.2.1 University Collaboration with MEMEX

Università Ca' Foscari Venezia is one of the leading universities in Italy in several topics like Economics, Management, Humanities, Literature, Languages and Science. The University, always devoted to the study of new technologies, is linked with 17 European public and private research centres, since 2005 with the foundation of the European Centre for Living Technology (ECLT) [1]. ECLT has been coordinator and partner in several FP6, FP7, H2020 and EuropeAid projects, where several groups are actively developing research collaborations both theoretical and methodological.

Our main focus in this collaboration will be clustering on Knowledge Graphs (KG) with the usage of Dominant Set [35].

Research and development activities carried out within the MEMEX-KG project focus to use the knowledge graph framework and related machine learning technologies to integrate services that can benefit the vulnerable communities targeted by the pilot program. Regarding the goal of the project, which is to integrate technological solutions to promote and enrich digital narratives to raise awareness of the challenges faced by target groups in the process of social integration in the EU, one important task is to provide meaningful context for the background cultural heritage and localized information in order to link the memories and experiences of these communities with the European Union's cultural reference points. In order to achieve this overall goal, part of the MEMEX-KG project focuses on a specific task, which can be expressed as a general machine learning problem: the static location of graph clustering.

Graphs are used to relate different elements together, this is conceptualised to aid in the creation of stories around memories, locations or objects. To this end we can utilise clustering to help the recommendation modality in order then to assist in the development of stories by end users. Clustering will be used to select nodes and links based on their potential connections based on the data provided by the users, such data could be connected due to period, content or other to enable users to create a narrative with meaningful content.

Clustering process is iterative in nature, so it gains a greater context, as more

[1] https://www.unive.it/pag/23664/

parts of the stories are created by the user the results is that the information provided are more specific to the user needs. The KG will research solutions by using dominant-set clustering [35] to select a set of nodes to which the user is interested in. The practice of grouping and combining information with a certain set of preferences is usually known in the literature as constrained clustering. Typically all the clusters are retrieved and then only the ones that contains the selected nodes are kept. An additional task of clustering is to consider navigation-based story creation. This adds an extra parameter for creating the clusters, i.e. the position of the user in the world. Therefore, when identifying relevant sets of nodes, it is important to consider their geographical distance as well as similarity.

Alternatively, smarter selection methods exist, like constrained dominant set [27]. Here the user can select dynamically the interesting nodes. The algorithm then will extract in one shot the relevant information, instead of computing an extensive search.

Finally, it would also be beneficial to guide the user in creating stories that progress across a space, as jumping back and forth will be a jarring and tiring experience to users, irrelevant of the excitement of the story. Stories can be easily revisited and the clustering re-performed to allow increased collaboration as new intangible elements are added to the graph, truly encapsulating the co-authoring of our shared heritage narrative [2].

1.3 Thesis outline

The choice of this H2020 project was mainly because of the technologies chosen for the project. Important for me was the fact that the purposes of the objectives were practical and noble. Studying arising technologies and new ways to implement them, would introduce me into the research world as well as applying on the field concepts of Artificial Intelligence that so far had been mainly theoretical.

This thesis begins with an introduction of all the preliminary concepts required to understand this work. In the next chapters we hence present the theoretical knowledge behind the work we performed. We explain concepts related to knowledge bases, graphs, knowledge graphs, multigraphs. Afterwards we focus on some models used to extract embeddings for the creation of similarity matrices. We have used this similarities to build simple weighted graphs that we have used for the main clustering algorithms: Dominant set [35], K-means [21], DBSCAN [16], Louvain [24] and Spectral clustering [51].
Dominant set will be presented extensively. The theory, general implementations and a focus on their contribution to knowledge graphs will be discussed. Moreover in chapter 3 we will present our extension to the algorithm.
In chapter 4 we present the structure of our datasets and the obtained results and

our analysis of them, focusing on their strengths and what should be improved.

The thesis ends with an overall recap of our work and final considerations.

Chapter 2

Background Knowledge

In this chapter we will see some basic concepts about graphs to then move into the concept of knowledge base and knowledge graphs (**2.1**). Then we will introduce the graph embedding techniques in **2.2** that can be used to build a pairwise similarity matrix. The concept of clustering will then be briefly explained (**2.3**), mainly focusing on graph clustering. In Subsec. **2.4** we will have a look at the most commonly used techniques for graph clustering and in **2.5** we present some local clustering methods. Finally in Subsec. **2.6** we are going to present the chosen metrics for the evaluation of such techniques.

2.1 Basic concepts to introduce the subject

2.1.1 Types of graphs

Definition 2.1 (Simple graph [9] [55]) A simple graph G is a pair $G = (V, E)$ where

- V is a finite set, called the *vertices* of G
- $E \subseteq V \times V$.

The graph edges could have **weights** to indicate the strength of the connection between the nodes.

Definition 2.2 (Directed graph [9] [55]) A directed graph (or digraph) have *edges* with direction, which indicate a one-way relationship. Formally, we can define a directed graph as a triple $D = (V, E, \Phi)$ where V and E are finite sets and Φ is a function with domain E and codomain $V \times V$. We call E the set of edges of the graph D and V the set of vertices of D.

Definition 2.3 (Undirected graph [9] [55]) An undirected graph instead have *edges* without direction, which indicate a two-way relationship. In other words is a set of *vertices* (or *nodes*) that are connected together by bidirectional *edges*.

Multigraph

We can also define a graph with multiple edges with the same nodes and we call it

Definition 2.4 (Multigraph [9] [55]) As for the simple graph, we can define directed and undirected multigraph, specifically:

- **directed multigraph** (edges without own identity) G: is an ordered pair $G := (V, A)$ where V is a set of nodes and A a multiset of ordered pairs of vertices, called directed edges or arcs.
- **undirected multigraph** (edges without own identity) G: is an ordered pair $G := (V, E)$ where V is a set of nodes and E a multiset of unordered pairs of vertices, called edges.

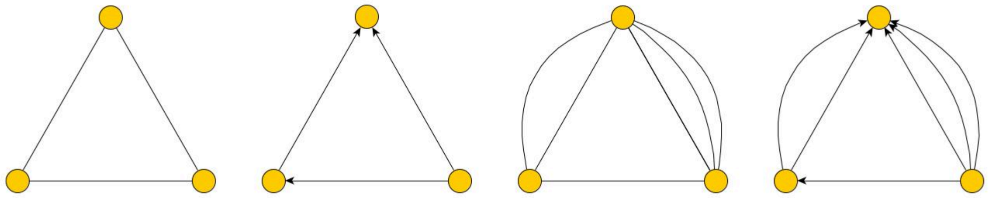


Figure 2.1: Example of graphs. From the left: Undirected graph, directed graph, undirected multigraph, directed multigraph.

Adjacency matrix

Definition 2.5 (Adjacency matrix) The adjacency matrix of a graph G is the $n \times n$ matrix $\mathbf{A}_G := (a_{ij})$, where a_{ij} is the number of edges joining vertices i and j. Can be described as:

$$a_{ij} = \begin{cases} k \text{ if } (v_i, v_j) \in E \\ 0 \text{ otherwise} \end{cases}$$

Using the adjacency matrix we are able to represent information in a graph. The non-zero entries indicated a edge between two nodes, whose weight is provided by the value of the entry. We could have non-zero diagonal elements of an adjacency matrix only if a node is connected to itself. Also the adjacency matrix of an undirected graph is symmetrical along the diagonal.

Cosine similarity matrix

A matrix of similarities can be constructed using the feature vectors. Projecting the vectors into the unit sphere throughout the Euclidean (L2) normalization and then performing their dot product gives us what it is commonly referred as the cosine similarity kernel [26]. Therefore the cosine similarity k is defined as:

$$k(x, y) = \frac{xy^T}{\|x\| \, \|y\|} \tag{2.1}$$

where the x and y variables are the rows of the feature vectors. This operation is the cosine of the angle of the points represented by the vectors.

2.1.2 Knowledge Base

The Knowledge Base (KB) is a database for managing information. Using this kind of databases makes easier the collection and organization of the knowledge in a particular area or a general one. Knowledge bases are used in computer science, in the development of expert systems and artificial intelligence algorithms [45]. Knowledge bases can be distinguish in the following:

- **Knowledge base of an expert system:** it collects and organizes the main knowledge into a given field of knowledge. For example, a medical, legal, technical knowledge base, etc. The knowledge of the human expert or of a group of experts on the subject is organized within it. The same information can then be used by non-expert users of the expert system [49].

- **Knowledge base of a logical agent [46]:** it has a collection of the main formulas and directives to allow the logic agent to move and make autonomous decisions in an external environment. It is the representation of the reality of the logical agent. Over time the agent assimilates experience and, through a process of inference, can modify the very content of the knowledge base.

Representation of knowledge: in the knowledge base, facts are presented in the form of formulas. Knowledge can be introduced into the knowledge base through different methods (declarative method, procedural method, neural network, etc.) and is expressed through a special language composed of symbols, syntax and semantics. For example, in the knowledge bases of the 70s-80s, knowledge was introduced by two primitive instructions: TELL (knowledge input) and ASK (knowledge query) [46].

2.1.3 Knowledge Graphs

The concept of knowledge graphs is strictly related to the one of knowledge base. The main difference is that it can be represented as the names express, by a graph. The idea of structure knowledge in a graph was proposed by Stokman and Vries in 1988 [47] , but only in 2012 the concept gained popularity when it was first launched by Google [14]. Between 1988 and 2012 Resource description framework (RDF)[1] and Web Ontology Language (OWL)[2] were released in turn, and became important standards of the Semantic Web [3].
The Knowledge graph contains a collection of interlinked descriptions of entities, concepts or events. Data is put in context via linking and this way provides a framework for data analysis. The KG descriptions have formal semantics that allow both people and computers to process them in an efficient and unambiguous manner, where entity descriptions contribute to one another providing context for easy interpretation.

Definition

Since the concept of a knowledge graph is still a new topic of interest, there is still not a wide-accepted formal definition. Some experts in the area have given some possible definitions of a knowledge graph:

Definition 2.6 (Färber.[3]) In the context of the Semantic Web we can define a knowledge graph $G = \{ \varepsilon$, R, F} where ε, R, and F are sets of entities, relations and facts. Facts represent knowledge with the form of triples (head, relation, tail) using the resource description framework (RDF).

Definition 2.7 (Ehrlinger and Woß [15]) A knowledge graph acquires and integrates information into an ontology and applies a reasoner to derive new knowledge.

Definition 2.8 (Wang et al.[39]) A knowledge graph is a multirelational graph composed of entities and relations which are regarded as nodes and different types of edges, respectively.

[1] https://www.w3.org/TR/1999/REC-rdf-syntax-19990222/
[2] http://w3.org/TR/owl-guide
[3] http://w3.org/standards/semanticweb

Definition 2.9 (Paulheim.[19]) A knowledge graph mainly describes real world entities and their interrelations, organized in a graph. It defines possible classes and relations of entities in a schema and allows for potentially interrelating arbitrary entities with each other. It also covers various topical domains.

2.1.4 MEMEX-KG

Knowledge Graphs (KG), or more familiar in Cultural Heritage as 'Semantic Graphs' are considered as ontologies of connected data. In contrast many KGs are more organically grown from a priori ontologies.

The definition just mentioned in the previous section helps describe the scope of operations and research in the context of the knowledge graph, and expresses the key issues to consider when constructing, analyzing, and applying these frameworks.

Regarding the activities carried out in the MEMEX project [2], these definitions reflect the development principles of MEMEX-KG: a flexible data structure that can integrate information about the European cultural heritage and the history of the target communities of the project, with clear and consistent insight. For these reasons, the construction follows a more formal and general definition related to graph theory.

General structure of the knowledge graph:

Given

- Set of E entities (nodes)
- Set of K relationships (edges)

Considering that

- Each entity $e \in E$ contains several attributes that vary from entity to entity;
- Each relationship $k \in K$ is directed and has a specific type encoding logical relationships among entities;
- Each attribute has a specific data type.

The resulting graph G(E,K) can therefore be defined as an attributed directed multi-graph.

At its core, a KG is a graph database and consists of a set of interconnected typed entities (nodes) and their attributes. The distinctive features of KGs lie in their special combination of knowledge representation structures, information management processes and search algorithms.

Following these definitions different Knowledge Graphs have been generated in the context of the MEMEX-KG, following a specific methodology for crawling the data from the open source web resources and encoding the necessary knowledge to be deployed in the different applications of the project within the graph. In chapter 4 the data acquisition methodology for generating the Knowledge Graphs will be presented along with the main datasets used. In section 4.2 the activities of research for the help to creation and enriching relevant stories and the development will described.

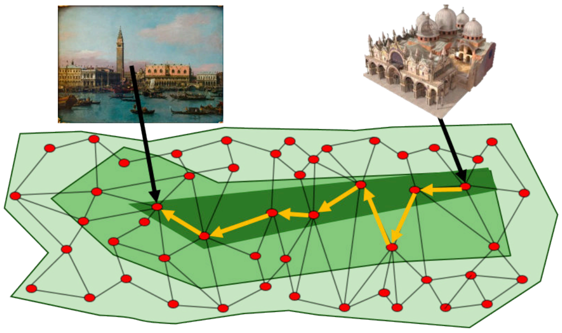


Figure 2.2: Example of how to utilize the MEMEX's Knowlege Graph (Image from [2])

2.2 Graph embedding techniques

In this section we will talk about some of the general machine learning techniques involving embeddings which will later on be used to build our similarity matrices.

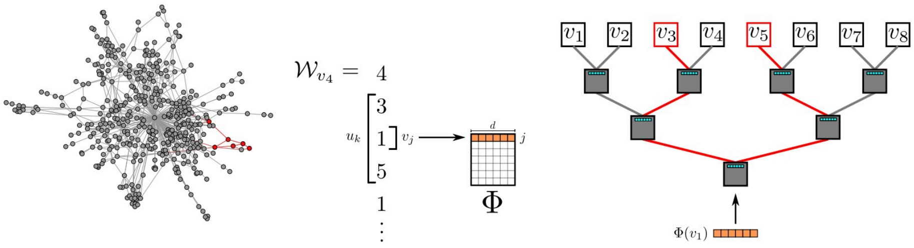
Definition 2.2.1 (Embedding) An embedding [1] is a relatively low-dimensional space that can be translated to high-dimensional vectors. Embeddings facilitate machine learning tasks on large inputs (i.e. sparse vectors representing words). Embeddings capture some of the semantics of the input by placing semantically similar inputs close together in the embedding space.

2.2.1 Topology Embeddings

DeepWalk

The idea of DeepWalk [38] is similar to Word2vec [29] [28], using the co-occurrence relationship between nodes in the graph to learn the vector representation of nodes. The key question is how to describe the co-occurrence relationship between nodes. The method given by DeepWalk is to use RandomWalk to sample nodes in the graph.

RandomWalk is a depth-first traversal algorithm that can repeatedly visit visited nodes. Given the starting node of the current visit, randomly sample a node from its neighbors as the next visit node, and repeat this process until the length of the visit sequence meets the preset condition.



(a) Random walk generation.

(b) Representation mapping.

(c) Hierarchical Softmax.

Figure 2.3: DeepWalk example. Image from [30]

After getting a sufficient number of node access sequences, it uses skip-gram model [30] for vector learning.

The DeepWalk algorithm mainly includes two steps. The first step is to sample the node sequence of random walks, and the second step is to learn the expression vector using skip-gram model Word2Vec. Briefly:

- Construct a homogeneous network, starting by sampling Random Walk separately from each node in the network to obtain locally associated training data;
- SkipGram training of sampled data, representing discrete network nodes as direct quantification, maximizing node co-realization, and using Hierarchical Softmax as a classifier for ultra-large-scale classification.

Node2Vec

Node2vec [18] is a graph method that combines DFS (Depth First Search) neighborhoods with BFS (Breadth First Search) neighborhoods. Simply put, it can be seen as an extension of deepwalk, which combines DFS and BFS random walk.

Node2vec still uses a random walk method to obtain the nearest neighbor sequence of a vertex, the difference is that node2vec uses a biased random walk.

Given the current vertex v, the probability x of accessing the next vertex v is

$$P(c_i = x | c_{i-1} = v) = \begin{cases} \frac{\pi_{vx}}{Z} & \text{if } i \in E \\ 0 & \text{otherwise} \end{cases} \tag{2.2}$$

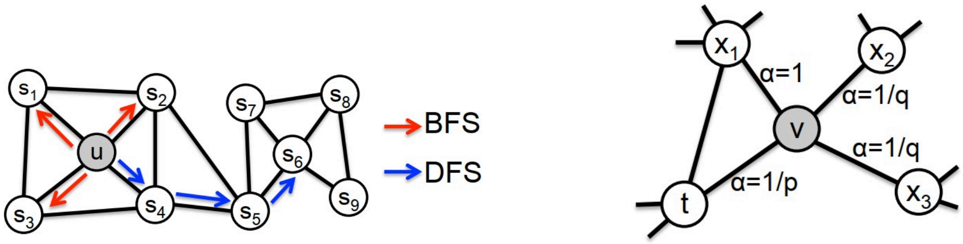
where π_{vx} is the probability of transition between vertex v and vertex x, with Z the normalization constant.

Node2vec introduces two hyper-parameters p and q to control the random walk strategy.

$$\alpha_{pq}(t, x) = \begin{cases} \frac{1}{p} & \text{if } d_{tx} = 0 \\ 1 = & \text{if } d_{tx} = 1 \\ \frac{1}{q} = & \text{if } d_{tx} = 2 \end{cases} \tag{2.3}$$

The impact of hyper-parameters p and q on walk-through strategies is discussed below.

- **p:** Parameter p controls the probability of repeated access to the vertes you have just visited. If p is higher, the probability of accessing the vertex you have just visited is lower and, conversely, higher.
- **q:** q Controls whether the walk is outward or inward, and if, $q > 1$ random walk tends to access visit and approach the t-close vertices (biased towards BFS). If $q < 1$ it tends to access vertes away from t (biased toward DFS).



(a) DFS and BFS

(b) Node2Vec

Figure 2.4: Node2Vec example. Image from [18]

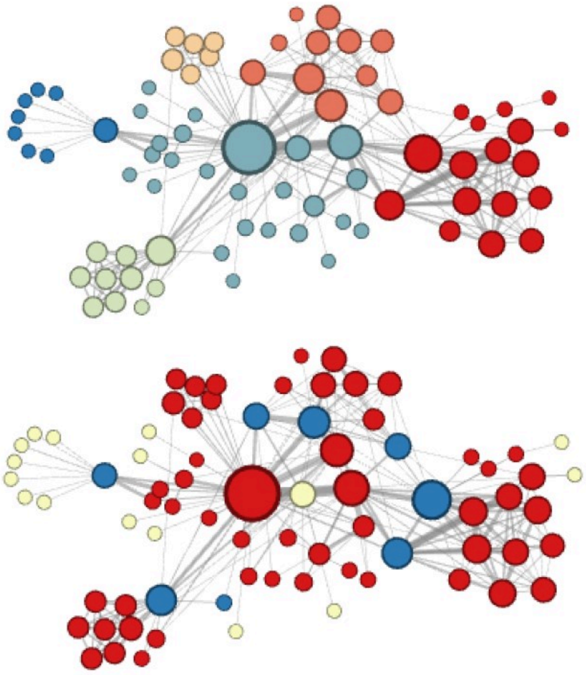


Figure 2.5: Node2Vec example. Image from [18]

Struc2Vec

Struc2Vec [40] defines vertices similarity from the perspective of spatial structural similarity. In fact, in some scenarios, two vertes that are not close neighbors may also have high similarities that cannot be captured. Struc2Vec is useful for this kind of scenarios. Struc2Vec's paper was presented at the 2017 KDD Conference [40].

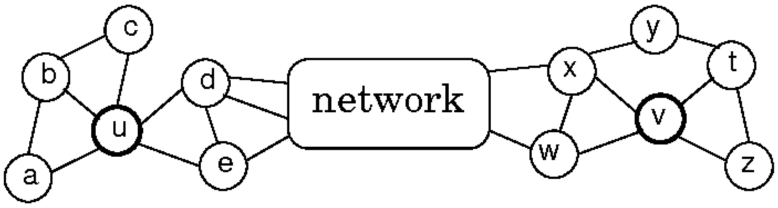


Figure 2.6: Struc2Vec example. Image taken from [40]

Usually when a vertex u and vertex v are not similar in a model based on near-neighbor similarity, the first observation that can be made is that they are not directly connected, and the second is that they do not share any neighbor vertex. In struc2vec's hypothesis, vertex u and vertex v are spatially similar. Their degrees are 5 and 4, respectively, connecting 3 and 2 triangular structures and through 2 vertices *(d, e; x, w)* is connected to the rest of the network.

Intuitively, vertices with the same degrees are structurally similar, and if their adjacent vertex points still have the same degrees, they are more similar.

2.2.2 Semantics embeddings

Word2Vec

One of the methods to construct efficiently word embeddings is Word2Vec [28] [29]. Many successful natural language processing tasks have utilized embeddings learned through the utilization of Word2Vec. The main ideas from the papers [28] [29] for the learning representations of words are the following:

- **Continuous Skip-gram.** The model considers words into a vector one at a time. Each word is scanned within a certain range before and after the current word in the same sentence. The ranges are n-grams, where an n-gram is a contiguous sequence of n items in a linguistic sequence.

- **Continuous Bag-of-Words.** This model predicts words based on the average of their vectors. Specifically the distributed representations of the surrounding words are combined in order to predict the word in the middle (which is the current one). For this model the order of the words is not important, since we take the average.

In practice Skip-gram has been shown to have good results since is able to positively score rare words or phrases, even when the size of the training dataset is relatively small. On the other size the computational time in order to train Continuous Bag-of-Words is way smaller than the skip-gram, and has a slightly better accuracy for the frequent words.

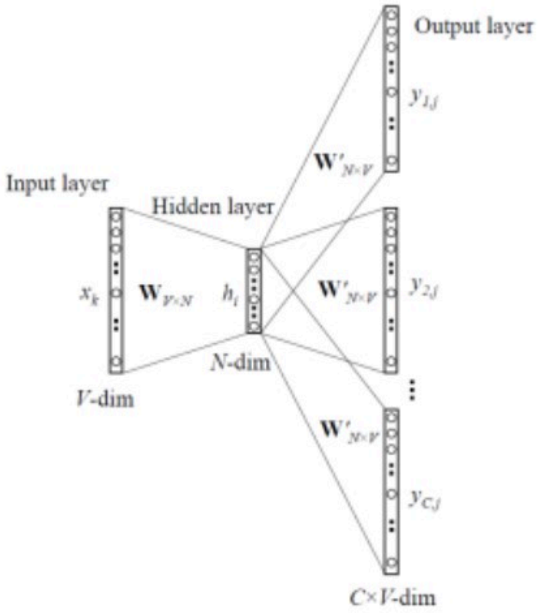


Figure 2.7: Skip-gram example. Image from [28]

FastText

Facebook research team in 2016 [7] proposed what's is call bag of tricks for efficient text classification.
FastText main idea is to incorporate words into the skip-gram model (sub-words). The final word embedding vector will then be the sum of all the n-grams of the starting word.
The model in general is very fast, simple and outperforms models that don't take into consideration the sub-word information. It also takes into consideration less common words since increases the probably that other n-grams is more likely.

2.2.3 Translational models

Translational models basically model graph relationships by interpreting them as translations in the embedding space. They have received a major number of attention, in the link prediction task [42]. Instead we have used them to retrieve the embeddings to build up our similarity matrix. In section **2.1.3** we have seen that with definition 2.1.6, a KG can be seen as a set of triples representing each *fact* in the graph.

The triplets consist of: (head entity, relationship, tail entity). Even though we won't be focusing on the evaluation of the link prediction task, since we are more interested in the embeddings themselves, we followed the ranking procedure proposed by the literature on benchmark datasets to know also the quality of the embeddings themselves.

For each triplet in the test set, the head entity is removed and replaced by each of the entities of the KG dictionary. The evaluation metrics were: Mean of Predicted Ranks (MRR) and Hits@10 (hits@N indicates the probability that the correct reasoning result appears in the first N results, which is similar to the recall rate of the knowledge reasoning algorithm).

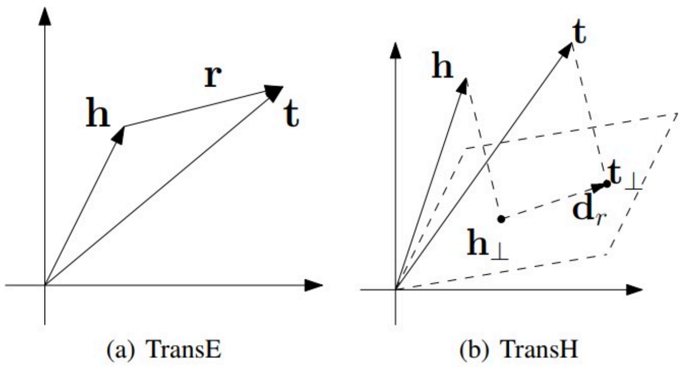


Figure 2.8: TransE and TransH. Image from [52]

TransE

In TransE [10] the relationships are represented as translations in the embedding space. The main idea is based on the fact that if the triple *(h,r,t)* is relevant and holds, then the embedding of the tail entity t and the head h should be close in the embedding space plus some vector that depends on the relationship r. The output given by this model are embeddings for entities and relations in $\mathbb{R}^k$ with where k is the hyperparameter of the dimentionality of the embedding.

The downside of this type of model is that is able to handle efficiently only one-to-one relations but is not able to take care efficiently of many-to-many, one-to-many and many-to-one relationships.

TransH

TransH [52] model differs from the previous one since it considers the relationship as a hyperplane where an operation of translation is computed.

In this way, the model is able to handle one-to-many, many-to-one, and many-to-many which TransE is not capable of, while using almost the same complexity.

TransR

In the TransR [25] model the general idea that differentiates itself from the previous two models is the fact that does not assume that the relation r and the entity e are in the same semantic space. The embedding construction is done in two steps:

- The entities are projected into the entity space and then projected to relation space according to their relation r;
- Once all entities have been projected, translations can be build.

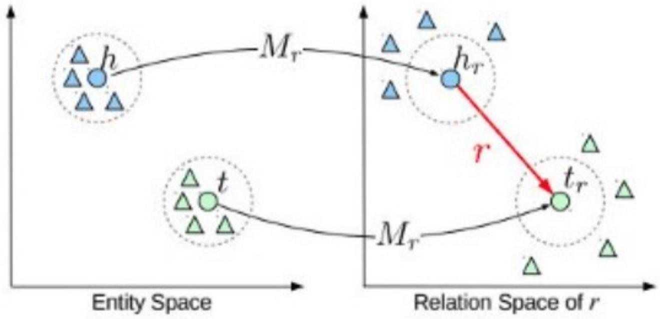


Figure 2.9: TransR. Image from [25]

2.3 Clustering

The purpose of this task is to allow the use of graph clustering techniques to create narratives around locations or objects. This task is challenging not only because of the need to merge the heterogeneous structure information (consisting of multiple types of nodes and edges) of the knowledge graph (KG), but also because of the need to consider heterogeneous attributes or content (for example, text, pictures, videos, etc.) associated with each node. In addition, this definition also poses many challenges in the effective use of data and the effective manipulation of graph-related structures that have high computing and storage costs.

2.3.1 A general view on Cluster

Part of the research process focused on the specific definition of the task, in fact clustering falls under multiple areas of research definitions. Clustering is a field of study which basic idea is rather simple and refers to the task aiming to discover, group and expose data into maximally coherent clusters in datasets [23].
Depending if the number of groups is already established we have:

- **Supervised clustering**: prior knowledge exists about the subject under study;

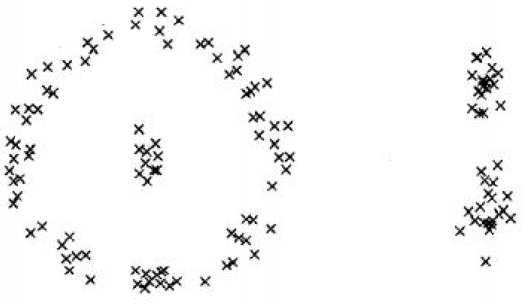


Figure 2.10: Example of clusters. Image from [36]

- **Unsupervised clustering**: no prior knowledge exists about the subject under study.

Therefore, part of our analysis focuses on using the information contained in the knowledge graph to find node partitions or subsets that may be relevant to the story generation/enriching process of target communities. The specific characteristics of the linguistic expression of stories make the process of formulating standards and benchmarks for system analysis very complicated. In this thesis, we consider static location graph clustering as an instance of unsupervised clustering.

For instance we cannot know the number of clusters that have to be found, but it can only be guessed playing we the hyper-parameters of the clustering technique.

Although there is no universal definition about clusters, generally all agree that a cluster must satisfy the following criterion:

- **Internal criterion**: objects within a cluster must be as similar as possible to each other;
- **External criterion**: objects outside a cluster must be as different as possible from those inside.

So the best clusters to be taken under consideration are the ones with a high internal homogeneity and a high external inhomogeneity.

Depending on the input provided to the clustering algorithm we have [33] [41]:

- **Feature-based clustering or central clustering**. The objects are represented by a vector of n features which, in turn, can be seen as a point in an n-dimensional space. This approach is limited by the fact that not everything can be represented through feature vectors;
- **Pairwise clustering or graph-based clustering**. The algorithm uses an adjacency (square and generally symmetrical) matrix between objects. Objects are represented by a graph not oriented but weighted, with as many vertices as objects and who among similar objects. Since here it doesn't

necessarily work on features vectors, pairwise clustering is a more general method and flexible than the previous one;

- **Hierarchical clustering.** Provides a representative hierarchical view of the various clusters. If the scale is fine we have as many singletons as points; if the scale is coarse we have gradually more numerous sets).

2.3.2 Clustering as a Graph-theoretic problem

When considering clustering as a graph-theoretic problem, we have the following:

- Set of n objects;
- n x n matrix A of pairwise similarities;
- A graph G, edge-weighted.

At this regard we can have several different similarity graphs. As stated in [51] and [31] we can have:

- **The binary similarity matrix.** This is the simplest similarity matrix, based on the topology of the graph. If an edge exists between node v_i and node v_j then v_{ij} equals to 1 and 0 otherwise;
- **The connection-based similarity matrix.** As for the previous one, this type of similarity matrix, simply corresponds to the number of edges between node v_i and node v_j of the graph G;
- **The ε-neighborhood graph.** Given a threshold ε and a set of data points $x_1, ..., x_n$, we connect the points whose pairwise distances are smaller than ε. This type of graph is typically considered as an unweighted graph since weighting the edges would not increase the quality of the information [51];
- **The k-nearest neighbor graph.** The similarity matrix is created by connecting the node v_i to node v_j if v_j is among the k-nearest neighbors of node v_i. The resultant graph will be a directed one. In order to make the graph undirected there are mainly two possible ways. We can have a *k-nearest neighbor graph* where we simply ignore the directions of the connection, so if we connected v_i and v_j with an edge if v_i is among the k-nearest neighbors of v_j or the opposite. The second option is the *mutual k-nearest neighbor graph* where the connection happens only if both v_i and v_j are among the k-nearest neighbors of each other;
- **The shortest-path connected graph.** The suggested approach measures the similarity between nodes in a graph by using a shortest path algorithm. Using the shortest path algorithm [12] the final similarity matrix have the following structure:

$$W_{ij} = 1/n \sum_{k \in P} w_k$$

where P is the set of the weights from i to j.

- **Embedding based similarity graph.** As discussed previously embeddings allow us to place similar inputs close together in the embedding space. The adjacency matrix of similarities can then be build up using cosine similarity [26].

2.4 General techniques for graph clustering

2.4.1 DBSCAN

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) [16] is a clustering method density based, since it connects regions of points with sufficiently high density.

DBSCAN estimates the density around each point by counting the number of points in a neighborhood ε specified by the user, and applies thresholds called *minPts* to identify the "core", "border" and "noise" points. In a second step, the core points are gathered in a cluster, if they are "density-reachable", that is, if there is a chain of core points in which each point falls within the eps-surrounding of the following. Finally the edge points are assigned to the clusters. The algorithm requires only the ε and *minPts* parameters.
We said that minPts identify three types of points which are:

- **A core point** is a point that has around it (within the eps distance) a number of other points at least equal to minSamples. A core point defines a cluster.

- **A border point** is a point around which there are fewer points than minSamples, but one of them is a core point. For this reason this border point is assigned to the cluster identified by the core point close to it.

- **A noise point** is a point around which there are fewer points than minSamples and none of these is a core point. This means that the noise point is farther than eps from any core point.

One of the characteristics of DBSCAN is to be able to manage clusters even if they are not spherical, even if, like many other clustering algorithms, it exploits a distance (which is therefore calculated on a circular, spherical or in general hyperspherical space).

2.4.2 K-means

K-means [21] is the most popular and simplest clustering algorithm. Given a set of n features vectors $x_1, ..., x_n$ and the number desired of cluster K, we have that:

- It randomly selects K points representative of the data, called centroids;
- It fixes the centroids and assigns all the remaining points to the closest centroid, using for example the notion of Euclidean distance;
- It fixes the assignments and recalculates the centroids on the new clusters just obtained, by computing the average of all the points of each cluster.
- Repeats the last two steps until there are not changes in the locations of the centroids or until the number of iterations set by the user is reached.

This algorithm final goal is the minimization of the objective function

$$F = \sum_{i \in clusters} \left(\sum_{j \in \text{elements of the i-th cluster}} \|x_j - u_i\|^2 \right) \tag{2.4}$$

where $\|x_j - u_i\|^2$ is the distance measure between the point x_j and the centroid u_i.

2.4.3 Spectral Clustering

Spectral clustering [51] is one of the most commonly known techniques for grouping and partitioning data in fields like machine learning, computer vision or signal processing. The object of clustering is to divide a given dataset into natural groups. Spectral clustering doesn't make any assumption on the form of the clusters and treats the problem as a graph partitioning one and has significant advantages in comparison to "classical algorithms" like k-means. One of the main advantages of spectral clustering it's its usage of linear algebra methods to solve efficiently the problem. In order for this algorithm to work we assume that the graph G = (V,E) is undirected and weighted. The resulting weighted adjacency matrix of the graph is the matrix W = (w_{ij}) with $i, j = 1, \ldots, n$. As G is undirected then $w_{ij} = w_{ji}$ in order to have a symmetric matrix. We need to define as well the degree of a vertex $v_i \in V$:

$$d_i = \sum_{j=1}^{n} w_{ij}$$

The degree matrix of the graph G is a diagonal matrix where the off-diagonal elements have value 0.

Graph Laplacian

One of the main characteristics of spectral clustering are the usage of graph Laplacian matrices. There are several forms of Laplacian matrices. We will describe them briefly.

Unnormalized graph Laplacian

The unnormalized Laplacian matrix L is defined by:

$$L = D - W$$

Key fact: the matrix L has the following important property defined as:

$$\forall \text{ vector f} \in R^n \text{ we have} f'Lf = 1/2 \sum_{i,j=1}^{n} w_{ij}(f_i - f_j)^2$$

Where n is the cardinality of v.

- L is symmetric and positive semi-definite: f'Lf $\geq$ 0;
- Smallest eigenvalue of L is 0;
- Corresponding eigenvector is 1;
- Eigenvalues are then: $0 = \lambda_1 \leq \ldots \leq \lambda_n$;
- *f'* is the derivative of *f*.

Studying the properties of the eigenvalues and eigenvectors of the laplacian of the given graph G many properties about the graph itself can be understood. Just to mention the relation between the eigenvalues and the structure of the graph we have that:

- The multiplicity of eigenvalue $\lambda_1 = 0$ is the number of connected components of the graph.
- eigenspace is spanned by the characteristic functions of these components (so all eigenvectors are piecewise constant)

Normalized graph Laplacians

There are two new versions of the laplacian, both normalized with different normalization criteria.
Row sum normalization or random walk normalization [51]:

$$L_{rw} = D^{-1}L \tag{2.5}$$

Symmetric normalization [51]:

$$L_{sym} = D^{-1/2}LD^{-1/2} \tag{2.6}$$

Both the L_{sym} and L_{rw} hold the properties of the unnormalized Laplacian matrix.

Algorithm

The algorithm's inputs are a similarity matrix $S \in \mathbb{R}^{n \times n}$ and a number k of clusters to construct. The algorithm returns k clusters. It follows these steps:

Algorithm 1 Normalized Spectral clustering [51]

- Construct a similarity graph and let W be its weighted adjacency matrix.
- Compute the normalized graph Laplacian L_{sym}.
- Embed data points in a low-dimensional space (spectral embedding) computing the k smallest eigenvectors $v_1, \ldots, v_k$ of L_{sym}.
- Let $V = [v_1, \ldots, v_k] \in \mathbb{R}^{n \times k}$.
- Form the matrix $U \in \mathbb{R}^{n \times k}$ from V by normalizing the row sums to have norm 1, that is:
$$u_{ij} = \frac{v_{ij}}{(\sum_k v_{ik}^2)^{1/2}}$$
- For $i = 1, \ldots, n$, let $y_i \in \mathbb{R}^k$ be the vector corresponding to the ith row of U.
- Cluster the points y_i with $i = 1, \ldots, n$ with the k-means algorithm into clusters $C_1, \ldots, C_k$.

2.4.4 Louvain community

Nowadays networks tend to have complex topologies with many interconnected entities on the scale of millions if not billions. For this purpose the extraction of useful information from this kind of networks is hugely necessary. The Louvain method [24] is used for the detection and extraction of the community structure from large networks throughout the optimization of the modularity. Modularity is a metric that offers a view on the coherence of the partitions. The modularity [37] of a partition is defined as a value that goes in range from -1 to 1 and it measures the density of links inside communities compared to links between communities. When the graph is weighted, modularity is defined by the following:

$$Q = \frac{1}{2m} \sum_{ij} [A_{ij} - \frac{k_i k_j}{2m}] \delta(c_i, c_j) \tag{2.7}$$

where A_{ij} represents the edge weights between node $g_i \in$ G and node $g_j \in$ G, k_i is the sum of the weights of the edges attaches to g_i, c_i is the community of the node g_i. The parameter m = $\frac{1}{2} \sum_{ij} A_{ij}$. Finally $\delta(u, v)$ is 1 if u = v and 0 otherwise.

Methodology

The objective of the Louvain algorithm is to maximize modularity. It achieves so by the iterating two phases. In the first step, a different community is assigned to each node. The second step consists in:

- For each node i ∈ G:
    - consider the neighbours j of i
    - evaluate the gain of modularity that would result by removing i from its community and by placing into community j.

The previous statement can be expressed mathematically by the following equation [24]:

$$\Delta Q = \left[\frac{\sum_{in} + k_{i,in}}{2m} - \left(\frac{\sum_{tot} + k_i}{2m}\right)^2\right] - \left[\frac{\sum_{in}}{2m} - \left(\frac{\sum_{tot}}{2m}\right)^2 - \left(\frac{k_i}{2m}\right)^2\right] \tag{2.8}$$

where $\sum_{in}$ is the sum of the edges inside c∈C, $\sum_{tot}$ is the sum of the edges incident to nodes in C, k_i is the degree of node i, $k_{i,in}$ is the sum of the weights of the edges from node i to the nodes in C, and m is the sum of all the edges in the graph. Once the value is calculated, i is placed into the community with the higher score of modularity. The process is applied until a local maximum is reached.

The second phase consists in building a new graph whose nodes are the communities found in the first phase. Links between nodes of the same community are now represented by self-loops. Links from multiple nodes in the same community to a node in a different community are represented by weighted edges between communities. Every time the second phase finishes, the first phase can be re-applied until all communities have been found. By construction the number of communities decreases each time. The advantages of this algorithm are its simplicity and implementation.

2.4.5 Dominant sets

Introduction

Usual clustering techniques make some strong assumptions:

- The clustering problem is the problem of the best partitioning of the data (by doing so excludes the possibility of overlapping clusterings)

- The affinity matrix is symmetric, in order to be able to compute the real eigenvectors and eigenvalues.

The dominant set approach [35], instead, takes more in consideration **what it is a cluster**, how to define the **internal criterion of similarity** and the **external criterion of similarity**.

Definition

The Dominant Set algorithm is a generalization of the maximum clique problem where the edges of the graph are weighted.

Definition 2.4.5.1 (A clique [4]) *A clique is a maximal complete subgraph where each couple of nodes is connected between them. Another explanation, is a subset where the introduction of a new node to the clique makes the clique incomplete.*

As we already discussed, the graph G is presented in the form of its adjacency matrix A with $a_{i,j} = \omega(i, j)$ where ω is the weight of the edge between node i and j.

If we consider a simple scenario in which the affinity matrix A is binary (containing 1 if two objects are similar and 0 otherwise). We therefore obtain an undirected and unweighted graph in which we have as many vertices as points and an edge only when two nodes are connected. In this situation a cluster is in fact exactly the maximal clique of the graph. Therefore the notion of a clique would be equivalent to the notion of a cluster in this thesis.

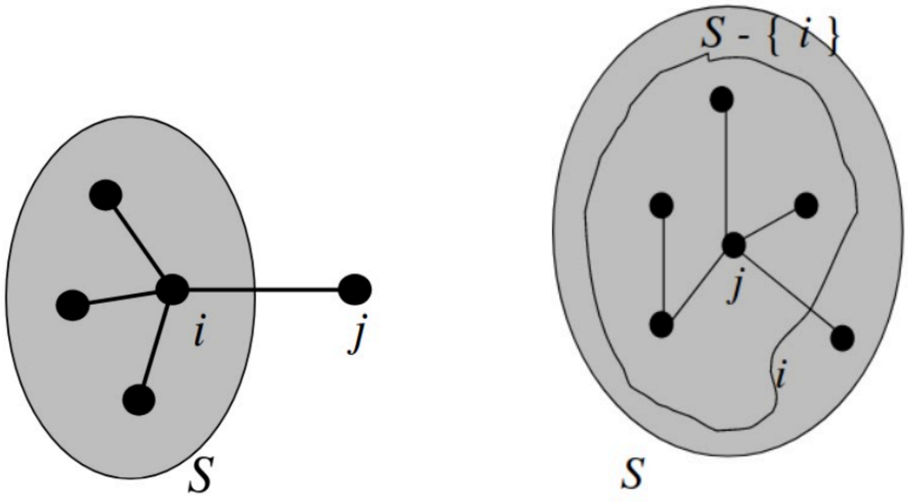
As anticipated in the introduction dominant sets try to answer to the question **what is it a cluster** and tries to prove measures of cohesiveness of a cluster and node participation.

Although there isn't a well defined definition of cluster, they always have to satisfy two conditions:

- **High internal homogeneity.** A cluster contains objects highly similar to each other.

- **High external inhomogeneity.** Different clusters are highly dissimilar between each other.

To define the dominant set we first need a couple of preparatory concepts. Let us consider an undirected and weighted graph G = (V, E), where the weights depend on the affinity matrix A.



(a) Measure of relative similarity

(b) Total weight of S

Figure 2.11: Notation. Images from [36]

Let us take a nonempty subset of S $\subseteq$ V such that the sum of weights of any subset of S is always positive ($W(T) > 0 \; \forall \; T \subset S$). We can then:

Definition 2.9 (Average weighted degree) [36] *The average weighted degree quantifies how much an element $i \in S$ is related to S by computing the affinities between the vertex i and the rest of the vertices of S (i. e. normalizing for the cardinality of S):*

$$awdegS(i) = \frac{1}{|S|} \sum_{j \in S} a_{ij} \tag{2.9}$$

Definition **2.8** gives, if positive, that the similarity between i and j is more relevant, if negative then the similarity between i and S is more prevalent (figure a 2.11).

Definition 2.10 (Measure of relative similarity) [36] *The measure of relative similarity φ, given $j \notin S$ quantifies if an element $i \in S$ is more linked to S or to a generic element j external to S with respect to the average similarity between i and its neighbors in S.*

$$\varphi_S(i, j) = a_{ij} - awdeg_S(i) \tag{2.10}$$

Definition 2.11 (Weight of an element) **[36]** *We can define the impact or the weight of a generic vertex i inside a set of vertices in S:*

$$w_s(i) = \left\{ \begin{array}{cc} 1 & if \quad |S| = 1 \\ \sum_{j \in S \setminus \{i\}} \varphi_{S\setminus\{i\}}(j,i) w_{S\setminus\{i\}}(j) & otherwise \end{array} \right\} \tag{2.11}$$

Intuitively, w$_S$(i) represents how similar i is with respect to the entities in S. More Formally, w$_S$(i) provides a measure of the overall (relative) similarity between the vertex i and S$\backslash\{i\}$ with respect to the overall similarity among the vertices of S$\backslash\{i\}$.

Finally the total weight of S is defined as:

$$W(S) = \sum_{i \in S} W_S(i) \tag{2.12}$$

W(S) gives a measure of the support that an object i receives from the objects in $W_{S\backslash\{i\}}$. When the value of W(S) in respect to i is positive then i is highly similar to $W_{S\backslash\{i\}}$ and should therefore be part of the set S.

We all this information we can now define Dominant set in this way:

Definition 2.12 (Dominant Set) **[35]** *A non-empty subset of vertices $S \subseteq V$ such that W(T) $>$ 0 for any non-empty $T \subseteq S$, is said to be a **dominant set** if:*

- *w_S(i) $>$ 0 $\forall i \in S$ (internal homogeneity)*
- *$w_{S\cup\{i\}}$(i) $<$ 0 $\forall i \notin S$ (external homogeneity)*

From the definition of dominant set we can notice how the two conditions are strictly related to the definition of a cluster. Informally we have that if i is extraneous to the set S the weight is negative, otherwise is positive. A dominant set is therefore a set of vertices that are maximally cohesive with each other.

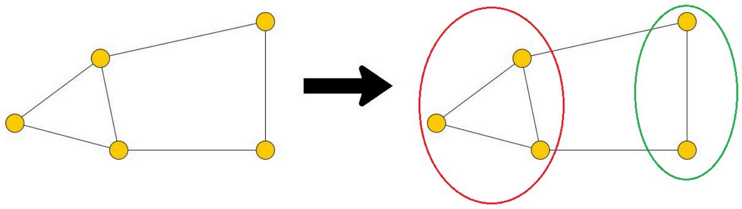


Figure 2.12: Example of Dominant Set.

Link to optimization theory

Since the definition dominant set is equivalent to the one of a cluster, in order to understand why it generalizes the maximal clique problem seen previously, let's remember that the cohesion of a cluster is measurable as

$$f(x) = x^T Ax \tag{2.13}$$

where A is the affinity function. It can then be demonstrated that if S is a dominant set, then its characteristic vector x^S belonging to the standard and n-dimensional simplex, is a strict local maximum of $x^T Ax$.

Dominant sets are in fact in one-to-one correspondence to (strict) local solutions of Standard Quadratic Program [8].
Given a symmetric matrix A, we can now formulate the clustering problem as the problem of finding the vector x that maximizes f. Therefore we have the following optimization problem:

$$\begin{array}{ll} \text{maximize} & f(x) = x'Ax \\ \text{subject to} & \text{x} \in \Delta \subset R^n \end{array}$$

where

$$\Delta = \{x \in R^n : x_i \geq 0 \wedge \sum x_i = 1 \forall i \in V \tag{2.14}$$

is the standard simplex of R^n.

We have that x is a strict local solution of the quadratic problem if the sum of all the components of the simplex is 1 and all its components are non-negative. It's important to define the support of a vector $\sigma(x)$ of $x \in \Delta$ which correspond to the index set of the positive components of x:

$$\sigma(x) = \{i \in V : x_i > 0\}$$

Theorem 1 (Dominant Set) [35] [43] If S is a dominant subset of vertices, then its weighted characteristic vector x^S is a strict local solution of program **2.4.5**. Conversely, if x^* is a strict local solution of program (1), then its support $\sigma = \sigma(x^*)$ is a dominant set.

By the theorem 1 we have that dominant sets correspond to strict local solutions of the quadratic program seen previously. In particular we have that its weighted characteristic vector x^S is a strict local solution and since x $\in \Delta$ belongs to the standard simplex, then the characteristic vector can be defined as:

Definition. 2.15 (weighted characteristic vector) [35] [43] A dominant set can be always found in a non-empty subset $S \subseteq V$. It admits a weighted characteristic vector $x^s \in \Delta$ if it has positive total weight $W(S)$, in which we can set:

$$x_i^S = \begin{cases} \frac{W_s(i)}{W(S)} & \text{if } i \in S \\ 0 & \text{otherwise} \end{cases} \tag{2.15}$$

By construction dominant set always admit a characteristic vector.
The results shown by theorem 1 are different and interesting since solutions are found by looking at the standard simplex. In contrary many other solutions focused more into the sphere. Some of the advantages of this approach are the following:

- the weighted characteristic vector can be used as a measure of the participation of the vertices in the cluster;
- the value returned by the objective function gives a natural way of defining the cohesiveness the cluster itself;
- avoids the utilization of negative numbers which are not relevant.

Link to Game theory

Dominant Set and evolutionary game theory [54] have a subtle marriage. This allow us to find dominant sets and optimize the solutions through the utilization of deterministic game dynamics. Game theory used as an extension of graph theory for the dominant set gives powerful results. The game theory will be shortly described topic for the thesis background.

Game theory can be described as the science of strategy, also referred as the optimal decision-making through the the study of mathematical models of rational strategies.

In this section we will see the part of game theory that is needed know to understand the functioning of dominant set with game theory. In particular we need to specify that the link between dominant set and the optimization theory discussed before, its possible only in the case of a symmetric matrix. The game theory is instead more flexible since it works also with asymmetric similarity matrices.
In this sense we can model the classical clustering problem into a new **non-cooperative clustering game** with the following properties:

- a finite number of players I = $\{1, \ldots, n\}$ with $n \geq 2$. The players must be at least two;

- each player has a finite number of said actions pure strategies, $S_i = \{1, \ldots, m_i\}$ with $m_i \geq 2$. Every player must have at least two pure strategies available;

- the Cartesian product of pure strategies forms the strategic profile of the game, S = $S_1 \times S_2 \times \cdots \times Sn$.

- a payoff function, π, which maps pure strategic profiles a real values, one for each player.

At this point we can represent a game in normal form as a triplet G = (I, S, π). Note that the function depends on the actions of all players, not just on the action of the individual agent; clearly every player wants to maximize their payoff.

Symmetric two-player games

In the special case where there are only 2 players, the payoffs can be represented by two square matrices, A and B, where:

- A contains the payoffs of the first player: $a_{ij} = \pi_1(i, j)$ with i $\in S_1$, j $\in S_2$

- B contains the payoffs of the second player: $b_{ij} = \pi_2(i, j)$ with with i $\in S_1$, j $\in S_2$.

Some historical examples of two-player games are:

- **Zero-sum games.** Studied deeply by Von Neumann, this are games in which, given a certain pure strategic profile, what player 1 gains is strictly equal to what player 2 loses. This means that the sum of their respective payoff matrices is equal to A + B = 0;

- **Symmetric games.** In this games each player's role can be swapped. A classic example is rock-scissors-paper. In particular this example is both symmetric and zero-sum. Formally we have that the transpose of B is equal to A: A = B^T;

- **Doubly-symmetric games**. In this games the player's payoff matrix is symmetric to itself, such that A = $A^T = B^T$.

Let's imagine that the i-th player plays rock-paper-scissor many times: if in most cases he chooses rock, the other player can understand his strategy and consequently use paper. In general it is therefore preferable to choose between the available pure strategies S_i based on a probability distribution. A **mixed strategy** is a probability distribution on the set of pure strategies, indicated by the vector $x_i \in \Delta$ with Δ once again the standard simplex such that:

$$\Delta = \{x_i \in R^n : x_i \geq 0 \wedge \sum x_i = 1 \forall i \in V\}$$

This vector is made up of as many components as there are strategies of player i, each indicating the probability that the corresponding pure strategy will be used by the player. This probability is always non-negative. The set of pure strategies with probability different than 0 forms the support of x_i. Each player has his own mixed strategy and the vector containing the mixed strategies of all n players is called the mixed strategy profile, x = $(x_1, \dots, x_n)$. The expected payoff is based on the probability that a certain pure strategy profile s (containing the pure strategy chosen by each player) is chosen when a mixed strategy profile x is played.
The probability that a pure strategy profile s is used, when a mixed strategy profile x is played, is given

$$x(s) = \prod_{i=1}^{n} x_{i_{s_i}} \tag{2.16}$$

Basically the payoff of a player i is therefore the sum, for all possible pure strategies, of the product of the probability of using strategy s and the payoff obtained using strategy s. The payoff of player i is therefore given by

$$u_i(x) = \sum_{s \in S} x(s) \pi_i(s) \tag{2.17}$$

We can now modify our definition of a game and define it with mixed strategies. We now have a triplet G = (I, Θ, u) where I is the players set, Θ is the mixed strategies space and u is the payoff function for mixed strategies. In the special case of two-player games, it is possible to represent the payoff function with a pair of matrices (A, B) where A (B) is the payoff matrix of player 1. We have then that the expected payoffs for both the players are

$$u_1 = x_1^T A x_2$$

and

$$u_2 = x_2^T A x_1$$

Linking back to dominant set, the stable mixed strategy is the one in which objects belonging to the same cluster are selected. In this way is possible to maximize the payoff of both players. In order to make the payoff coincide with the definition of a cluster, the desired condition is to have what is called a **Nash equilibrium**. A Nash equilibrium is a configuration of mixed strategies (one per player) such that no player has an incentive to change their strategy. Nash equilibrium assumes that other players don't change their strategies either. Formally we have that the vector x = $(x_1, \ldots, x_n)$ is a Nash equilibrium if it is the best answer to itself.

$$y_1^T A x_2 \leq x_1^T A x_2 \qquad y_2^T A x_1 \leq x_2^T A x_1 \qquad \forall (y_1, y_2) \in (\Delta \times \Delta).$$

In the case in which $x_1 = x_2$, then we have what is called **symmetric Nash Equilibrium**. The previous two conditions can be rewritten as:

$$y^T A x \leq x^T A x$$

This condition can be seen as a definition of internal homogeneity for the clusters, and can be reformulated as:

$$\begin{cases} (Ax)_i = x^T A x & \text{with } i \in \sigma(x) \\ (Ax)_i \leq x^T A x & \text{with } i \notin \sigma(x) \end{cases}$$

The Nash equilibrium is said to be an **Evolutionary Stable Strategy (ESS)** if it satisfies the following condition as well:

$$y^T A x = x^T A x \implies x^T A y < x^T A x \qquad \forall y \in \Delta \setminus \{x\}$$

(Theorem 4) Let A be the similarity matrix of a clustering problem instance and let Γ be the corresponding clustering game. If S is a dominant set of A then its characteristic vector x^S (see, Eq. (2)) is an ESS of Γ. Conversely, if x is an ESS of Γ, then S = $\sigma(x)$ is a dominant set of A, provided that $(Ax)_i \neq x^T Ax \; \forall i \notin S$.

In conclusion we can say that:

- the notion of evolutionary stable strategies is equivalent to the notion of dominant set and vice versa;
- ESSs are strictly in correspondence to strict local solutions of constrained quadratic forms.

Clustering using dominant sets

In this section we will focus on how does dominant set perform clustering. The main advantage that dominant set has it's the fact that the coding part of it is very simple. Also, being strictly related to game theory, it should not come as a surprise that the method used are from a game theoretic notion.

Replicator dynamics

Replication dynamics are a class of dynamic systems studied in the context of evolutionary game theory, a discipline born by J. M. Smith [53].
An example of replicator used in context developmental game theory to apply the Darwinian concept of fitness. The idea is simple: strategies whose payoff is greater than the average are destined to spread to the population; strategies the whose payoff is below average are instead destined to disappear over time.
Let's consider a very large (ideally infinite) population belonging to the same species competing for a particular set of limited resources, such as food, water, etc. Suppose as well that each individual is prescheduled to play a particular pure strategy and the payoff represents reproductive success of the species. This type of situation can be seen as a game in which two players are randomly chosen from the population in order to compete with each other. The player who wins contributes to the survival of the species. The evolution of the dynamics, due to the principle of natural selection, will show that the stronger individuals (the ones that adopted the winning strategy), will tend to dominate the weaker individuals, and as a direct consequent, the weaker will eventually be extincted.

Let $x(t) \in \Delta$ be the vector representing the state of the population at time t, then $x_i(t)$ is the amount of the population playing the pure strategy i at time t. If we have a fixed individual A from the population programmed to play strategy i and we randomly choose an opponent from the population, and let them participate in game G. The expected payoff received by A would be equal to:

$$u(e^i, x)$$

Instead if A is chosen randomly, then the expected payoff would be:

$$x_i u(e^i, x)$$

Finally, the average payoff of the population is:

$$u(x, x) = \sum_{i=1}^{k} x_i u(e^i, x)$$

In the theory of evolutionary games, two assumptions are made:

- The population plays iteratively generation after generation;
- The strategies are not necessary rational, but based on the natural selection where the stronger survives.

A general class of evolution equations can be described by

$$\dot{x}_i = x_i g_i(x), \forall i, i = 1, \ldots, n$$

where $g_i(x)$ specifies the replication ratio of the pure strategies i, with g= $(g_1, \dots, g_n)$ (usually) regular. The regularity $g(x)^T x = 0$ guarantees the uniqueness of the solution between the simplex Δ.
If we consider

$$g_i(x) = u(e^i, x) - u(x, x)$$

then we get the standard replicator dynamics equations

$$\dot{x}_i = [u(e^i, x) - u(x, x)]x_i$$

The replicator dynamics used for the optimization comes from the payoff-monotonic game where as the name **monotonic** suggest, prioritizes the rate of replication of the strategies with a higher payoff, penalizing the ones with a lower one. We can call payoff-monotonic a regular selection dynamics when:

$$g_i(x) > g_j(x) \Leftrightarrow (Ax)_i > (Ax)_j, \forall x \in \Delta \wedge i, j \in V$$

As a direct consequence of the previous, a subclass of payoff-monotonic game dynamics is given by:

$$\dot{x}_i = x_i \left[\phi((Ax)_i) - \sum_{j \in V} x_j \phi((Ax)_j) \right] \tag{2.18}$$

Here the $\phi((Ax))$ is indeed the monotonic function for the value Ax. It is easy to prove that when ϕ coincides with the identity function then (number) becomes

$$\dot{x}_i = x_i[(Ax)_i - x^T Ax] \tag{2.19}$$

Linking back to Darwin's principle of natural selection we have that:

$$\frac{\dot{x}_i}{x_i} \propto \text{payoff of pure strategy i - average population payoff}$$

which yields:

$$\dot{x}_i = [u(e^i, x) - u(x, x)]x_i = x_i[(Ax)_i - x^T Ax] \tag{2.20}$$

Theorem x (Nachbar, 1990; Taylor and Jonker, 1978) [36] *A point $x \in \Delta$ is a Nash equilibrium if and only if x is the limit point of a replicator dynamics trajectory starting from the interior of Δ.*
Furthermore, if $x \in \Delta$ is an ESS, then it is an asymptotically stable equilibrium point for the replicator dynamics.

The ESS are particularly interesting because they can utilize the asymptotic stability condition. However in this case, unfortunately, we cannot affirm that there is a one-to-one correspondence between ESS and asymptotically stable points. Instead, an interesting result that links ESS balances to doubly symmetrical games. We assume that the payoff matrix A is doubly symmetric ($A = A^T$). Thanks to this assumption we can derive the following useful properties:

Fundamental Theorem of Natural Selection (Losert and Akin, 1983) [36] *For any doubly symmetric game, the average population payoff*

$$f(x) = x^T Ax$$

is strictly increasing along any non-constant trajectory of replicator dynamics, meaning that $\frac{df(x(t))}{dt} \geq 0 \; \forall t \geq 0$, *with equality if and only if* $x(t)$ *is a stationary point.*

Characterization of ESS's (Hofbauer and Sigmund, 1988) *For any doubly symmetric game with payoff matrix A, the following statements are equivalent:*

- $x \in \Delta^{ESS}$
- $x \in \Delta$ *is a strict local maximizer of* $f(x) = x^T Ax$ *over the standard simplex* Δ.
- $x \in \Delta$ *is asymptotically stable in the replicator dynamics.*

There are a few ways to implement the extraction of dominant sets. One way is to apply the Runge-Kutta method, which is a well known numerical iterative method used for optimization. However this is not easiest way. A straigth fordward way is the utilization of discrete-time first order replicator equation derived from 2.4.5 and equal to:

$$x_i(t+1) = x_i(t)\frac{A(x_i) - \alpha x_i(t)}{x(t)'(A - \alpha I)x(t)} \tag{2.21}$$

where α is a parameter that is has been discovered to be related to the quadratic program and the maximum clique problem by E.G. Straus and T.S.Motzkin [32]. This intuition was latter on extended by Jagota and Pelillo (1995) where they found out how the maximal clique problem could be seen as a local maximizer of the objective function f $\in \Delta$. Finally is has been proved by Bomze in 1997 [8] that by setting the parameter $\alpha \in]0,1[$ the solution to the maximal clique problem of the graph G is given by the fact that the locals maximizers become characteristic vectors, where each vector corresponds to a clique. Reminding ourself that the dominant set is a generalization of the maximal clique problem and setting $\alpha = 0$ then we have that dominant sets can be found by the implementation of the following formula:

$$x_i(t+1) = x_i(t)\frac{A(x(t))_i}{x(t)^T Ax(t)} \tag{2.22}$$

Finding multiple clusters

The main strategy used in this work thesis to extract dominant sets is what has been called as **peeling-off** strategy. The main idea is to iteratively extract dominant dominant sets and removing the vertices from the problem. In this way every cluster will not have repeated nodes. This implementation is obviously not optimal, since starting from the second iteration dominant sets found are not clusters of the original problem. However in practice this implementation has been used many times with promising results and with lack of overlapping clusters.

Extentions to Dominant Set

The concept of Dominant set has been utilized in several different contexts. We will now introduce them briefly in order to then introduce more in detail one of its extensions utilized in this work of thesis. In the security field some interesting utilizations are for example anomaly detection for videos and in the internet (Hamid, Dacier, Pham, Thonnard) [20]. Even in the medical field they have been proved to work well for brain activity analysis and 3D ultrasound registration (Banerjee, Adamos) [6].

Finally in computer vision they've been utilized for the detection of conversational groups in images and sequences (Sebastiano Vascon & M. Pelillo) [50], for Image geo-localization (Eyasu Zemene & M.Pelillo) [56], Person re-identification and Multi-target tracking (Y. Tariku) [48] and constrained image segmentation (E.Zemene, L. Tesfaye & M. Pelillo) [27].
The idea of Hierarchical clustering [34] has been further extended in 2016 by E. Zemene and Pelillo [27].

2.4.6 Comparison of the state of the art techniques

All of the chosen clustering techniques have its own pros and cons. Here we will resume them.

- **DBSCAN**
  - It does not require you to specify the number of clusters in advance.
  - Works well with arbitrary shape groups.
  - Is robust for outliers and their detection.
  - In some cases, determining an appropriate neighborhood distance (eps) is not easy and requires knowledge of the domain.

  - If the clusters are very different in terms of density within the cluster, DBSCAN is not suitable for defining clusters.
  - The characteristics of the clusters are defined by the combination of the eps-minPts parameters. Since we pass an eps-minPts combination to the algorithm, it is not possible to generalize well to clusters with very different densities.

- **K-Means**
  - Fast
  - Not very useful on anisotropic data
  - Embarrassingly parallel
  - The user has to specify k (the number of clusters) in the beginning
  - only handles numerical data
  - assumes that we deal with spherical clusters and that each cluster has roughly equal numbers of observations

- **Spectral Clustering**
  - Elegant, and well-founded mathematically
  - Works quite well when relations are approximately transitive (like similarity)
  - Very noisy datasets cause problems since “Informative” eigenvectors need not to be in the top few and performance can drop suddenly from good to terrible
  - Expensive for very large datasets since computing eigenvectors is the bottleneck

- **Louvain Community**
  - Scalability (performs faster on huge graphs than other methods)
  - Simple to code
  - Iterative process can hide small communities found during intermediate phases. The result may be a coarse-grained high level representation of communities, which may not have the granularity needed for analysis. Hopefully, the nature of the algorithm makes it simple to save intermediate phases’ results so we can analyze different communities structures at different levels
  - Heuristic used to initialize phases and find local maximums can lead to not reproducible and not always optimized results. But this is the same with all data algorithms relying on heuristic (K-Means for instance)

Analysis of Dominant set strengths and weaknesses

As we have analysed Dominant Set have a long history of successful implementations as well as being relatively fast to code and to extract. In this thesis we will be focusing on their behaviour in knowledge graphs with the implementation of plain dominant set clustering and constrained dominant set. We are interest to see their behaviour in comparison to several other clustering techniques.

We can summarize Dominant set properties in the following:

- **Separation of structure and noise.** In situations where there is a large amount of noisy data, Dominant Set is able to retrieve coherent clusters;
- **Local solutions.** As discussed, dominant sets don't have to look for global solutions, since they can be extracted by looking at local solutions;
- **Clustering generalization.** Dominant set can be easily generalized to
    - Hypergraph clustering;
    - Hierarchical clustering;
    - Constrain clustering;
- **Adjacency matrix.** Dominant set can be used with either directed, undirected graphs, positive weights or negative ones. All of this is possible since it does not make any assumption on the structure of the adjacency matrix being used.
- **Number of clusters.** In contrast to several other algorithms like K-means or even Spectral clustering, dominant set does not require an initial parameter indicating the number of desired clusters, since it extracts them all in a sequential way;
- **Ranking.** Dominant sets allows the ranking of the cluster's elements according to their centrality to the cluster itself;
- **Theory.** Dominant Sets formulation have proved to be strictly connected its theoretical results;
- **Deterministic.** Replicator dynamics used in dominant set are a deterministic game dynamic. This gives us a guaranteed of the clusters found each time with fixed parameters, without the need of wasting computational time;
- **Parameters.** Dominant set needs only a small number of parameters as input, in particular we have the distance and cutoff. The **cut-off** is the threshold value in which below it, values will be considered null. In this way they no longer belong to the cluster.

On the other side of the medal some negative examples can be found as well:

- The fact of using a deterministic game dynamic, can be also seen as a negative of this algorithm, since if the quality of a cluster is small, then that same result will be carried over;

- Dominant set, although having a small number of parameters, proved to be extremely sensible to them, returning in some scenarios, consistent different results with their alteration;

- Depending on the strategy used to find dominant sets, we can have some advantages and disadvantages. If the strategy used is the peeling-off strategy, we have that starting from the second iteration dominant sets found are not clusters of the original problem. Otherwise if the strategy used considers the possibility of having overlapping clusters, then more significant results can be found expanding however the computational time for the extraction of the clusters and the utilization of a pruning techniques to clean-up the resulting clusters.

We can indeed confirm that the dominant set framework is a really powerful one, that has proved itself to be relatively flexible. All this things considered make us curious and to a certain degree, confident about how should behave in the knowledge graph domain.

2.5 Graph querying

All the previous examined clustering techniques work on the full graph. However it is often impractical and meaningless to perform the clustering of all the nodes. Instead becomes a more challenging and rewarding problem to be able to detect the communities given as a seed/set of nodes. Here is where local clustering [22] comes to help and its extremely useful when working on huge datasets where clustering can definitely become an expensive computational problem. The problem can be easily formulated as finding the subset $S \in V$, where S is the set of all the overlapping communities that contain the seed $s \in S$.

Personalized PageRank

The most commonly used technique for local clustering is certainly the Personalized PageRank [5]. Here once again random walk with restart it's the main core of the algorithm. Starting from the seeds in the set S it considers the random walk $X_0, \ldots, X_n$. In each step it has probability to move from node u to node v equals to

$$\alpha \frac{A_{u,v}}{d_u} \tag{2.23}$$

where $\alpha \in (0, 1)$ and d is the directed edge of u. The probability of the restart is given by $1 - \alpha$. Then the walk starts at s and with probability α it continues to a random neighbor the the current node. In mathematical terms, the PR of a node is the unique stationary measure of the distribution p for the Markov chain (X_t).

$$p(v) = (1 - \alpha)Ax + \alpha E \tag{2.24}$$

The vector p is known as the PPR associated with the seed S.

Constrained Dominant Set clustering

The idea is the following; instead of finding all generic sets, we provide the algorithm a subset $S \subset V$ of vertices and we want the system to find the dominant sets that contains them.

Now the quadratic programs for this dominant set problem becomes:

$$\max_{x \in \Delta} f_S^{\alpha}(x) = x' \left(A - \alpha \hat{I}_S \right) x \quad x \in \mathbb{R}^n \tag{2.25}$$

with $\alpha > 0$ and where I_S is the diagonal matrix whose elements are set to 1 in correspondence to the vertices outside the subset S and zero otherwise

$$I_S = \begin{pmatrix} 0 & 0 \\ 0 & I_{n-k} \end{pmatrix}$$

If α is sufficiently large:

$$\alpha > \lambda_{max} \left(A_{V \setminus S} \right)$$

then all local solutions of the maximization problem will have support containing at least one element of S. With this extension of the algorithm it should be possible to retrieve local clusters that might help in the story enriching/creation for the KG.

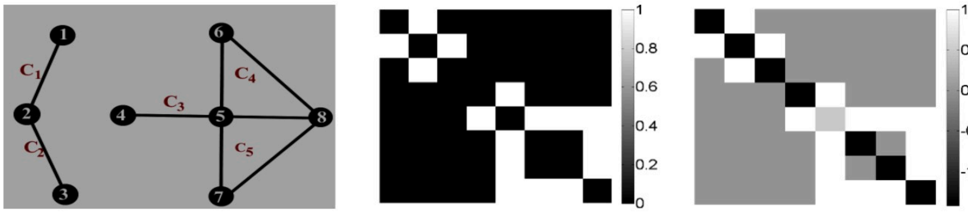


Figure 2.13: Example of CDS where the node 5 is chosen as seed node. Image from [27]

2.6 Metrics

In this section we present some of the metrics that we used to analyze the goodness of a cluster.

Community Modularity

The measure of performance we will use for the community detection is the modularity [37]. Modularity measures the strength of the division of a network into sub-groups. A network with high modularity has dense intra-connections (within sub-groups) and sparse inter-connections (between different groups).

The modularity of a partition is a scalar value between -1 and 1 that measures the density of links inside communities as compared to links between communities. In the case of weighted networks (weighted networks are networks that have weights on their links, such as the number of communications between two mobile phone users), it is defined as:

$$Q = 1/2m \sum_{i,j} [A_{i,j} - k_i k_j / 2m] \delta(c_i, c_j) \tag{2.26}$$

Where $A_{i,j}$ represents the edge weight, k_i k_j are the sum of the weights of the edges connected to nodes i and j; m is the total sum of all the edge weights in the graph; c_i and c_j are the communities of the nodes. Finally δ is the Kronecker delta that equals to one if i and j belong to the same community and 0 otherwise.

Davies-Bouldin index

When trying to measure the separation of clusters the Davies-Bouldin index [13] comes to our help, taking into consideration the average similarity of the clusters. The similarity here is obtained by calculating the distance between clusters with the size of the clusters themselves.

Silhouette Coefficient

The Silhouette Coefficient [44] is useful when the ground truth is not available. The Silhouette Coefficient score, similar to Community Modularity, is a scalar value between -1 and 1, where the higher stands for good defined clusters, the lower bad ones.

$$s = \frac{b - a}{max(a, b)} \tag{2.27}$$

Where:

- **a:** mean distance between an element $e_i \in E$ of the class E and all the other points $e \in E$.

- **b:** mean distance between an element $e_i \in E$ of the class E and all other points in the nearest cluster.

Calinski-Harabasz index

Another metric for unsupervised clustering is the Calinski-Harabasz index [11]. The C-H index measures the ratio of the sum of squared distances. In this way we obtain the ration of the between-clusters and within clusters.

Chapter 3

Dominant Sets for Knowledge Graphs

Considering that Dominant set has been successfully used in various scenarios regarding graph theory, we want to extend it, in order to apply it to the concept of KG defined in 2.1.
In specific in this thesis work we are trying to find coherent communities starting from the graph structure and then adding on top of them the other specifications of the knowledge graph definition. This work will be a preliminary step in order to explore the concept of KG in all its aspects, starting from each one singularly until all of them are covered. The whole idea is to be able to apply clustering techniques in such structures directly, without the need of the preliminary steps.

Considering the criterions that we discussed about in section 2.3, generally well-behaved clusters can be defined as the ones with a high internal homogeneity and a high external heterogeneity. When proceeding in defining a methodology for solving the unsupervised clustering problem stated above we defined limitations and challenges to be faced:

- **KG Structure:** information stored in the knowledge graph is generated considering several heterogeneous sources. This creates problems for the interpretability of the results and the consistency of the data. Several types can be combined with the very flexible KG design and with more information, increases as well the complexity of the data analysis and methods considered to solve this problem;

- **KG Representations:** In order to effectively extract relevant information from knowledge graphs, methods of feature learning were considered. However, each different strategies for creating useful and meaningful representations comes with its own disadvantages, related to limitation of the model in question and interpretability of the data itself;

- **Concept of Similarity:** The clustering task defines the criterion to group objects according to certain specific concept of similarity. The choice of this criterion is not trivial, since its choice influences the possible various strategies to acquire the feature vectors. Defining the concept of similarity becomes even harder with the flexible and heterogeneous structure of the knowledge graph;

- **Complexity of the task:** In the state of the art the number of examples where clustering techniques are applied to knowledge graphs are limited. Generally, the focus for this type of structures is related to their correct implementation on classic machine learning problems related to graphs. In addition, trying to enhance storytelling throughout clustering techniques increases the complexity of evaluating models and reaches some of the limitations of the current machine learning models.

Considering the above limitations in the task formulation of the problem research activities of the MEMEX-KG project, began with the studying and review of literature on graphs, graph clustering and knowledge graph embedding techniques for our experiments in order to define a structured theoretical background.

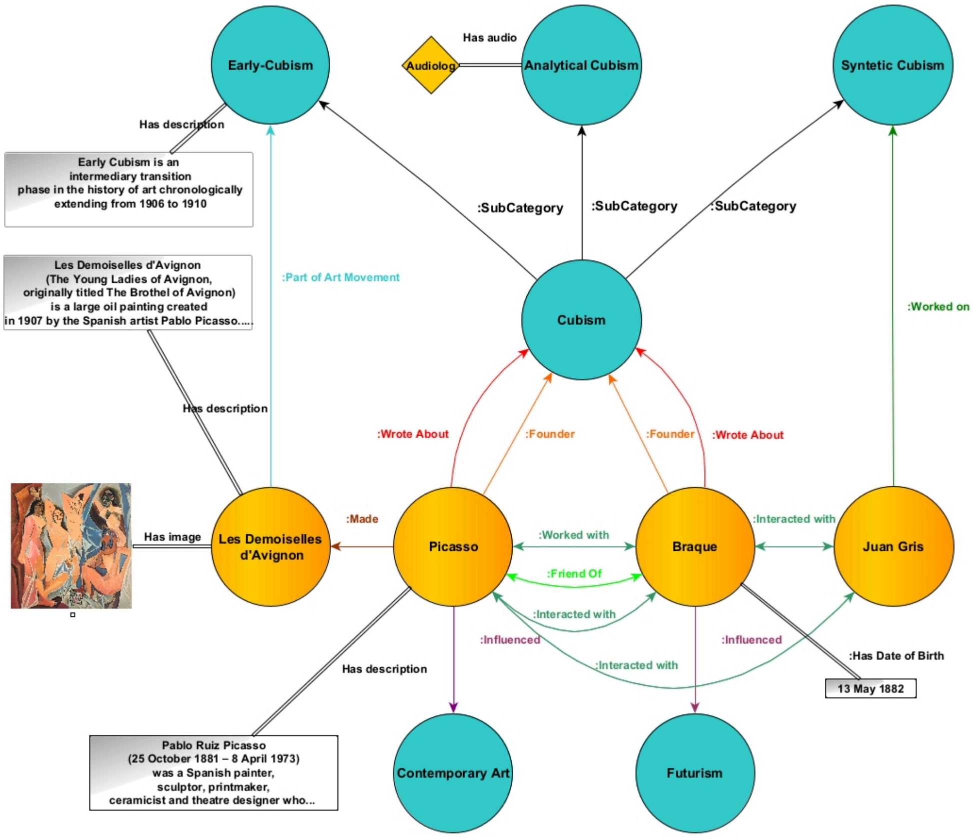

Early-Cubism
Has audio
Audiolog
Analytical Cubism
Syntetic Cubism
Has description
Early Cubism is an intermediary transition phase in the history of art chronologically extending from 1906 to 1910
:SubCategory
:SubCategory
:SubCategory
Les Demoiselles d'Avignon (The Young Ladies of Avignon, originally titled The Brothel of Avignon) is a large oil painting created in 1907 by the Spanish artist Pablo Picasso.....
:Part of Art Movement
Cubism
:Worked on
Has description
:Wrote About
:Founder
:Founder
:Wrote About
Has image
Les Demoiselles d'Avignon
:Made
Picasso
:Worked with
Braque
:Interacted with
Juan Gris
:Friend Of
:Interacted with
:Influenced
:Influenced
:Interacted with
Has description
:Has Date of Birth
13 May 1882
Pablo Ruiz Picasso (25 October 1881 – 8 April 1973) was a Spanish painter, sculptor, printmaker, ceramicist and theatre designer who...
Contemporary Art
Futurism

Figure 3.1: Cultural Heritage Knowledge Graph example

Starting from the definition of our KG 2.1.4 we have that a KG can be defined as a directed multigraph [9] [55]. Figure 3.1 shows a simple KG that we build to describe the problem, where the blue entities are "*Knowledge*" entities and the orange ones refer to "*places or people*". We can clearly see how many types of information is being represented by this data structure. In fact, as we remarked in section 2.1.3, an entity of the graph can have several types of information. Here we can see that the entity that refers to *Les Demoiselles d'Avignon*[17] contains an image of the art work and a description about the work, while at the same time the *Picasso* entity is storing only a description. We can also have many other types of information, like the *date of birth* stored in the *Braque* entity, or an *audio file* stored directly into the *Analytical Cubism* entity.

On top of all this kinds of data, we also have information in the edges. For example we can see that *Picasso* made the *Les Demoiselles d'Avignon* which is related to the *Early-Cubism* [17] period. Another example could be the people that interacted with *Braque*, which are *Picasso* and *Juan Gris*, but if we want to check on *Braque's friends*, then we will only have *Picasso*. All of this can be easily described as a multiview of the KG, and in figure 3.2 we can observe it.

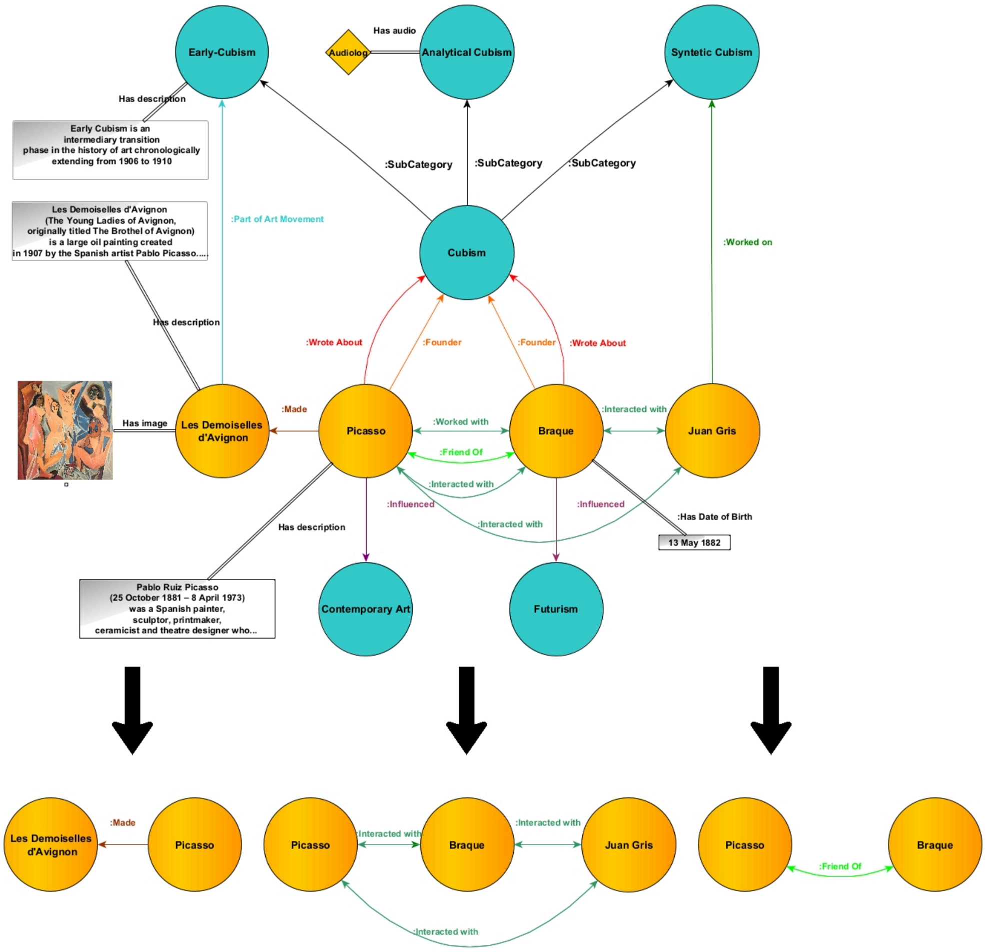


Figure 3.2: Multiview of a KG

Since Dominant set has never been used before in the context of Knowledge Graphs, but has been proved to work very well on graph structures we will try to extend the algorithm in order to explore its behaviour in the KG field.

We will begin our investigation starting from the transformation of a KG to a simple graph. In subsection 2.3.2 we discussed about some general types of similarity matrices. For example, the simplest case would be to have a binary similarity matrix A of the graph G, directed or undirected, simply considering that *1* means that two entities are connected and *0* otherwise.

Therefore, in order to try to capture more information about the amount of edges that a KG has, we can also consider the transformation of a KG into a un/directed weighted graph G, by considering the number of edges $e \in E$ of each node as the weight $w \in W$ of G, like we show in picture 3.3. In the undirected case, we would take the maximum number of connections between two entities.

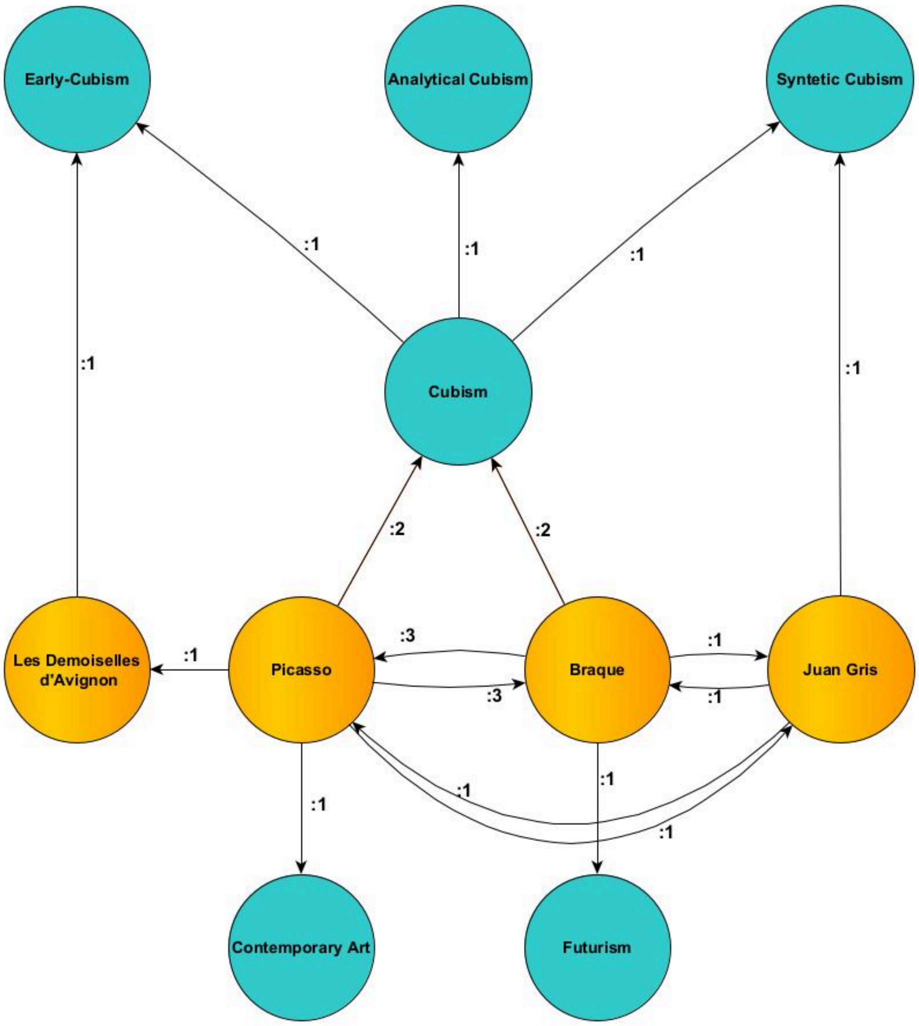


Figure 3.3: KG transformed into a simple weighted directed graph, considering the number of edges of each node as the weight

However we can clearly see that lots of information stored in the KG is being lost considering these approaches. We, therefore, thought about generating different similarity matrices, each one related to a specific aspect of a KG. The adjacency matrix of similarities can be build up as we said in 2.3.2 from the embeddings of each aspect of the KG. We considered the topology, the descriptions and the translational embeddings and we built of the matrices of similarities using the cosine similarity [26] that we have seen in section 2.1.1. In figure 3.4 the starting idea can be observed.

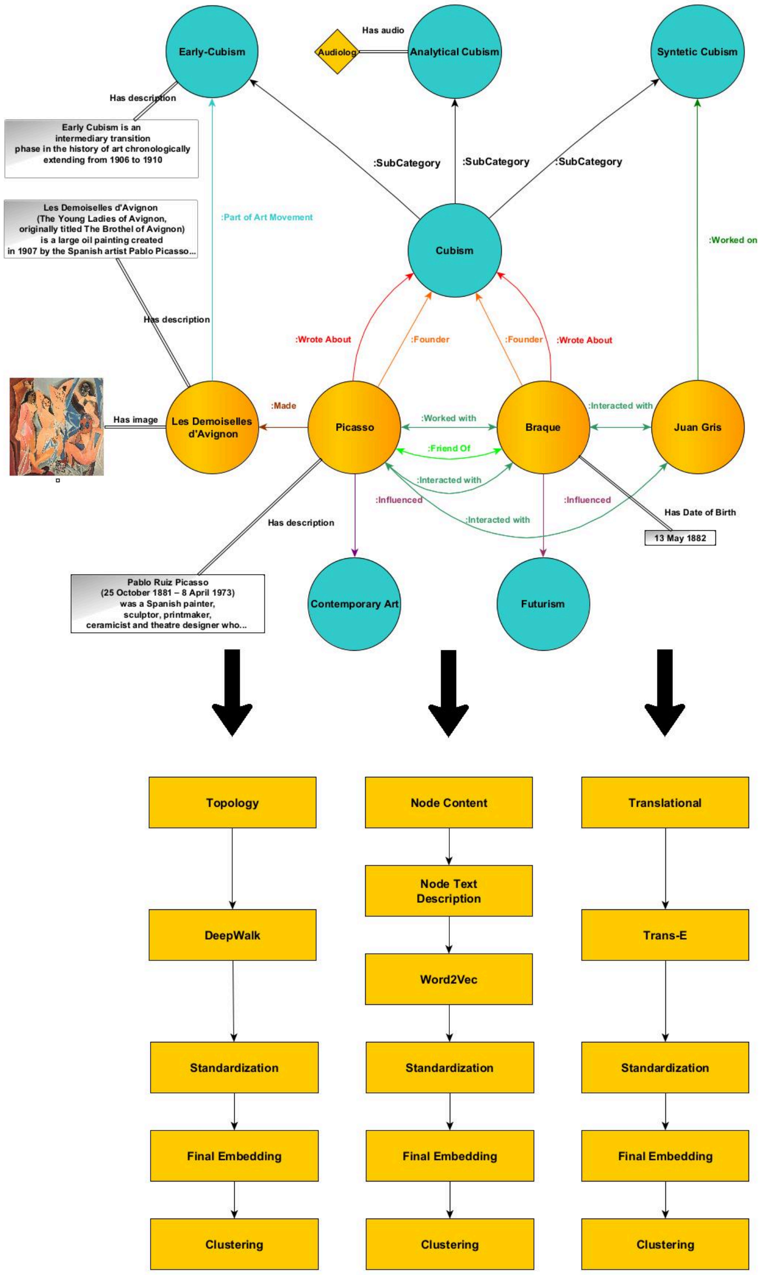


Figure 3.4: General idea for building new similarity matrices for KG and Dominant Set

Moreover in Picture 3.5 we extended further the general idea, by applying the concatenation of the different types of embeddings.

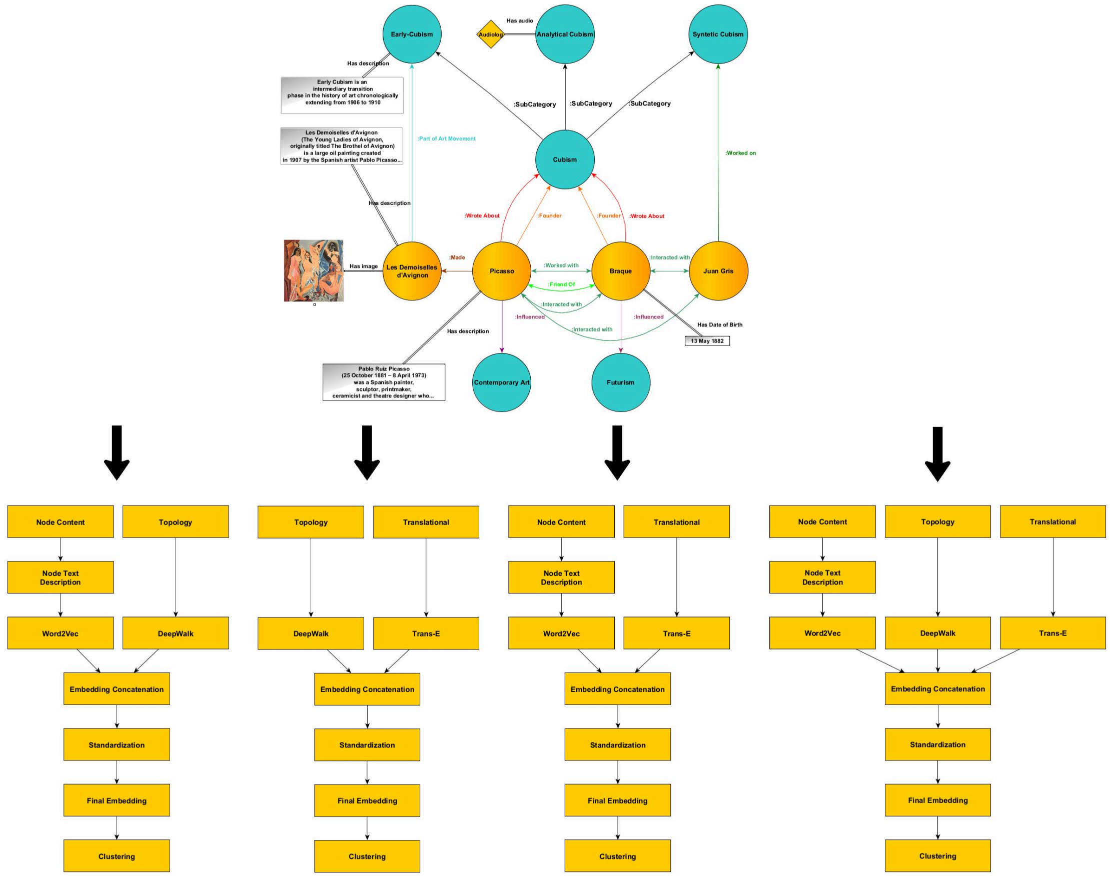


Figure 3.5: Extended idea for building new similarity matrices for KG and Dominant Set

Now our general approach for extracting Dominant sets that we have seen in 2.22

$$x_i(t+1) = x_i(t)\frac{A(x(t))_i}{x(t)^T Ax(t)} \tag{3.1}$$

can now be summarized as:

Definition. 3.2 (Dominant Set for Knowledge Graphs) A dominant set can be always extracted by:

$$x_i(t+1) = x_i(t)\frac{A_{emb}(x(t))_i}{x(t)^T A_{emb}x(t)} \tag{3.2}$$

Where A_{emb} the similarity matrix that we obtained through the embeddings calculation and concatenation.

3.1 Approaches for transforming the KG

Mainly four methodologies to build the matrix of similarity for Dominant set have been considered.

3.1.1 Structural clustering

The first method is to cluster KGs based on the simple graph's topology/structure information. In this case we are considering the matrices of similarities obtained squeezing the KG into simple graphs like we have seen in figure 3.3. More formally we will have that our new graph is an undirected weighted graph G = *(V,E,W)* where W is the set of weigths $w \in W$. In the binary case w will be just *1s* and *0s*, while instead in the weighted case $w_{ij} = \sum_{k \in P} e_k$ where $P \subseteq E$ is the subset of edges e that go from v_i to v_j.

3.1.2 Topology - Semantics - Translational

The second approach consists into performing clustering based on graph topology embedding. From this aspect we start out from the previously built simple graph and then we apply the topology algorithms that we have seen section 2.2.1 in order to extract the feature vectors. Therefore we build our graph G = *(V,E,W)* using the adjacency matrix returned by the cosine similarity kernel 2.1.1.

The third approach for clustering is performed based on the textual descriptions stored in the node entities. We proceeded extracting them and for each description, after the removal of the stop words, we simply computed sum of the embeddings of each word and then averaging by the number of words in the description. All of this in order to get a simple semantic representation of the description. After this preliminary step, we built our graph G using the same metodology as for the topology.

Finally the fourth step consists in extracting the graph embeddings, where heterogeneous edge types can be considered. Then we proceeded applying the cosine similarity 2.1.1 and building the matrix of similarities. The general approach is the one seen in figure 3.4.

3.1.3 Embeddings concatenation

This approach comes as a natural progression of the previous step, where the embeddings extracted have been concatenated in order to try to investigate if moving the data could gives some good results in the KG's field, specifically for the Cultural Heritage content.
Starting out from the previously extracted embeddings, we proceeded with their concatenation. Then we build our graph G = *(V,E,W)* using the adjacency matrix

returned by the cosine similarity kernel 2.1.1 in order to perform the clustering as seen in figure 3.5.

These three macro strategies were chosen to solve the limitations on clustering on KG that we mentioned at the beggining of this chapter, in order to try to expand the scope of our research activities.
We focus on the research using techniques in the literature to indicate whether the knowledge graph can benefit from the clustering task on top of them, and we will consider how our approach behaves as well.

3.1.4 Graph querying

Another extention to Dominant set will be to consider its behaviour by performing the graph query approach. In fact as seen in figure 3.2, one could easily exploit the information about the edges to perform graph query. In fact many times KG have been utilized for the link prediction task [42], however no information about the entities is utilized in this way. In order to try to explore this path we will implement the Constrained Dominant Set algorithm [27], and try to improved it for the KG task. We will start looking into the concept of graph querying, in order to check if valid and coherent information could be used for enriching stories creation.

Chapter 4

Application and Results

In this chapter we will discuss the datasets that we will utilize with our proposed methods. We will briefly present the preprocessing steps that we accomplished. Therefore we will analyze some of the problematics that might occur.

4.1 Preprocessing and Datasets

The structure of the Neo4j MEMEX-KG database can handle different types of entities and relations. For each of the pilot cities of the MEMEX project(Barcelona, Paris and Lisbon) we created a dataset, crawling data with the help of an ingestion tool. The crawler utilized is a custom ingestion tool that has been developed to handle CH data from heterogeneous sources: Wikidata [1], Europeana[2] and Mapillary[3]. In particular we focused in the analysis of all the CH objects belonging to each city utilizing the information retrieved from wikidata.

Briefly the ingestion method works as follows.
For the selected city it looks for the Wikidata items that have GPS coordinates within at preset range. These represent the starting point of our Places where related meta-data is downloaded. From each of the identified places, we then search associated nodes based on all relations, this new set of nodes we refer to as Knowledge. This could be for example, the painter of an art work. We repeat this step searching for new relationships a predefined number of times referred to as hops. For each of the pilot cities we considered both 2-Hops and 3-Hops.
After the crawling each dataset presented 3 type of nodes:

- **Place:** Physical CH places;
- **Knowledge:** Intrinsic information about the node;
- **WPI (Wikipedia Property Id):** Storing information about the nodes, used by the ingestion method.

[1] https://www.wikidata.org/wiki/Wikidata:Main_Page
[2] https://www.europeana.eu/en
[3] https://www.mapillary.com/

The first thing to do was to remove the WPI nodes from the Neo4j dataset, since they where not relevant for task problem, but just for the ingestion part.
At this point some preliminary analysis on the datasets have been done. In particular we have the following

Dataset	**Hops**	**Entities (nodes)**	**Relations (edges)**	**Types of relations**
Barcelona	2	7908	26705	170
Barcelona	3	15594	43934	394
Lisbon	2	2703	7412	132
Lisbon	3	9923	19900	317
Paris	2	29594	97750	239
Paris	3	62211	192036	562

Table 4.1: Datasets Information

The average node degree and density for each of the datasets is:

Dataset	**Hops**	**Average Degree**	**Density**
Barcelona	2	3.377	0.001
Barcelona	3	2.817	0.0001
Lisbon	2	2.742	0.002
Lisbon	3	2.005	0.0001
Paris	2	3.303	0.001
Paris	3	3.086	0.0001

Table 4.2: Datasets Average Degree and Density

As we can notice since we are dealing with large graphs and their scale increases as the number of hops considered increases. Evaluating the ratio between the actual number of connections in the networks and the potential number of connections resulted in low densities, it reflects the fact that the structures are weakly connected.
We also examined the top 10 relationships of the KG as well as the nodes with the most incoming and outgoing relations.

TOP 10 NODES WITH INCOMING CONNECTIONS				
DATASET	**NAME**	**TYPE**	**OUT**	**IN**
B2H				
	Spain	[’Place’]	0	5231
	Barcelona	[’Place’]	0	3899
	building	[’Knowledge’]	0	1310
	public art in Barcelona	[’Knowledge’]	0	1172
	sculpture	[’Knowledge’]	0	1018
	Cultural Asset of Local Interest	[’Knowledge’]	0	866
	Cultural Asset part of the architectural heritage of Catalonia	[’Knowledge’]	0	493
	masia	[’Knowledge’]	0	300
	vernacular architecture	[’Knowledge’]	0	288
	Art Nouveau	[’Knowledge’]	0	281
B3H				
	Spain	[’Place’]	0	5938
	Barcelona	[’Place’]	0	4442
	building	[’Knowledge’]	18	1350
	public art in Barcelona	[’Knowledge’]	6	1169
	sculpture	[’Knowledge’]	8	1021
	Cultural Asset of Local Interest	[’Knowledge’]	0	897
	Cultural Asset part of the architectural heritage of Catalonia	[’Knowledge’]	4	508
	human	[’Knowledge’]	0	448
	male	[’Knowledge’]	0	395
	Catalan	[’Place’]	0	327
L2H				
	Portugal	[’Place’]	0	1374
	cultural heritage	[’Knowledge’]	0	622
	Lisbon	[’Place’]	0	357
	building	[’Knowledge’]	0	177
	Immovable Cultural Heritage of Public Interest	[’Knowledge’]	0	162
	Santa Maria Maior	[’Place’]	0	144
	Included in protected site	[’Knowledge’]	0	136
	heritage without legal protection	[’Knowledge’]	0	119
	Santo António (Lisbon)	[’Place’]	0	78
	church building	[’Knowledge’]	0	72
L3H				
	Portugal	[’Place’]	174	1646
	cultural heritage	[’Knowledge’]	6	622
	Lisbon	[’Place’]	0	502
	building	[’Knowledge’]	0	183
	Santa Maria Maior	[’Place’]	0	166
	Immovable Cultural Heritage of Public Interest	[’Knowledge’]	6	156
	Included in protected site	[’Knowledge’]	0	142
	human	[’Knowledge’]	0	134
	Wikimedia category	[’Knowledge’]	0	131
	male	[’Knowledge’]	0	117
P2H				
	France	[’Place’]	0	17048
	Paris	[’Place’]	0	3269
	RATP	[’Knowledge’]	0	2431
	bus stop	[’Knowledge’]	0	2153
	street	[’Knowledge’]	0	1806
	registered historic monument	[’Knowledge’]	0	1621
	fountain	[’Knowledge’]	0	1494
	Smovengo	[’Knowledge’]	0	1193
	Vélib’ Métropole	[’Knowledge’]	0	1193
	bicycle-sharing station	[’Knowledge’]	0	1193

Table 4.3: Datasets Top 10 Entities with incoming connections

TOP 10 NODES WITH OUTGOING CONNECTIONS				
DATASET	**NAME**	**TYPE**	**OUT**	**IN**
B2H				
	Institut Ramon Llull	['Place']	33	0
	Cathedral of the Holy Cross and Saint Eulalia	['Place']	31	0
	Passeig de Gràcia	['Place']	29	0
	Gran Enciclopèdia Catalana	['Place']	26	0
	Fundació Jaume Bofill	['Place']	25	0
	Desolation	['Place']	25	0
	Centre de Cultura Contemporània de Barcelona	['Place']	22	0
	Casa Batlló	['Place']	21	0
	Gran Teatre del Liceu	['Place']	21	0
	Park Güell	['Place']	21	0
B3H				
	United States of America	['Place']	389	18
	Winston Churchill	['Knowledge']	188	1
	United Arab Emirates	['Place']	129	2
	Nelson Mandela	['Knowledge']	126	1
	Amsterdam	['Place']	116	1
	Ludwig van Beethoven	['Knowledge']	116	1
	FC Barcelona	['Place']	112	3
	IBM	['Knowledge']	106	1
	Spanish National Research Council	['Place']	104	7
	Norman Foster	['Knowledge']	104	1
L2H				
	European route E80	['Place']	75	0
	UEFA Euro 2004	['Place']	37	0
	Kingdom of Portugal	['Place']	28	0
	Santa Apolónia Station	['Place']	21	0
	Monastery of São Vicente de Fora	['Place']	20	1
	National Library of Portugal	['Place']	19	1
	Cais do Sodré railway station	['Place']	19	0
	Lisbon Cathedral	['Place']	18	0
	Colégio Militar	['Place']	17	0
	Church of Santa Engrácia	['Place']	16	0
L3H				
	Nigeria	['Place']	677	2
	Iran	['Place']	284	2
	Norway	['Place']	251	5
	Peru	['Place']	251	2
	Thailand	['Place']	246	2
	Czech Republic	['Place']	206	7
	Algeria	['Place']	204	2
	Ivory Coast	['Place']	194	2
	Ministry of Foreign Affairs of France	['Knowledge']	183	1
	Portugal	['Place']	174	1646
P2H				
	Roseraie de Bagatelle	['Place']	222	0
	Institut de chimie	['Place']	89	0
	The Triumph of the Republic	['Place']	88	0
	Court of Appeal of Paris	['Place']	64	0
	Orphanet	['Place']	53	0
	Supinfo	['Place']	52	0
	The Flight into Egypt	['Place']	44	0
	Le Génie de la Liberté	['Place']	40	0
	Wikidata Sandbox 2	['Place']	37	0
	Fame Fighting	['Place']	36	0

Table 4.4: Datasets Top 10 Entities with outgoing connections

TOP 10 RELATIONSHIPS		
DATASET	NAME	TYPE
B2H		
	instance_of	6424
	located_in_the_administrative_territorial_entity	5423
	country	5230
	heritage_designation	2679
	architectural_style	1326
	material_used	532
	architect	499
	part_of	372
	creator	326
	location	214
B3H		
	instance_of	7913
	located_in_the_administrative_territorial_entity	5676
	country	5628
	heritage_designation	2722
	architectural_style	1363
	subclass_of	1007
	occupation	890
	award_received	853
	topics_main_category	747
	languages_spoken_written_or_signed	680
L2H		
	instance_of	2378
	country	1320
	located_in_the_administrative_territorial_entity	1235
	heritage_designation	654
	location	288
	operator	99
	part_of	95
	located_in_time_zone	73
	applies_to_jurisdiction	70
	adjacent_station	70
L3H		
	instance_of	3206
	country	1637
	located_in_the_administrative_territorial_entity	1403
	language_used	914
	heritage_designation	661
	subclass_of	599
	contains_administrative_territorial_entity	593
	member_of	553
	topics_main_category	501
	described_by_source	490
P2H		
	located_in_the_administrative_territorial_entity	19324
	instance_of	19322
	country	16908
	located_on_street	7404
	shares_border_with	6155
	operator	3769

	named_after	3674
	heritage_designation	3536
	part_of	2019
	depicts	1455

Table 4.5: Datasets Top 10 Relationship types

We can see that some of the relations could surely introduce some bias in our knowledge graph, since they are highly connecting every entity of the graph, without really adding meaningful information. Also some entity nodes like the pilot cities themselves don't add up much information and, since the crawling data starts basically from them, the fact that they have many incoming connections its justified, yet not very useful. However removing this entities and relations would surely cause data fragmentation, since the average connectivity is around 3 for each dataset. Therefore this would end up in entities that are not anymore connected to the main graph, that will have to be removed if they end up without any connection. In our cluster analysis this will probably result in many more small clusters, specially for the topology analysis. Having taking care of this considerations, we proceeded to create some sub-graphs where we removed:

- **Nodes:** Spain, Barcelona, Portugal, Lisbon, France, Paris;
- **Relations:** instance_of, country, located_in_the_administrative_territorial_entity.

After the removal of the relations and the nodes, we clean up the data, removing the isolated nodes. In the following table we can resume the reduced datasets statistics.

Dataset	Hops	Entities (nodes)	Relations (edges)	Types of relations
Barcelona	2	5799	9449	167
Barcelona	3	13669	23739	391
Lisbon	2	1957	2374	129
Lisbon	3	9278	13080	314
Paris	2	25483	41818	236

Table 4.6: Reduced Datasets information

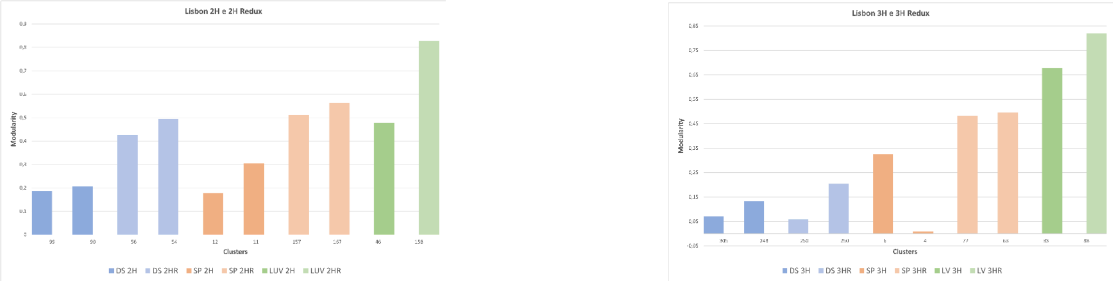
4.2 Results analysis clustering

In this section we are going to explore our results starting from a general overview of the topology and the behaviour of the clustering techniques on the data. For a fluid flow in both the presentation and analysis, we will mainly focus on the modularity metric discussed in section **2.6**. Other metrics will be consulted accordingly when discussing about the concept of clustering in the KG specific context.

4.2.1 Topology Clustering results

Let us start from our discussion about clustering as a graph theoretic approach. Like we discussed in section 3.1.1 we started out considering as our similarity matrix, the binary adjacency matrix of the graph in its undirected representation. However we immediately switched to the weighted representation since results where not satisfactory and no relevant information could be retrieved from the binary form.

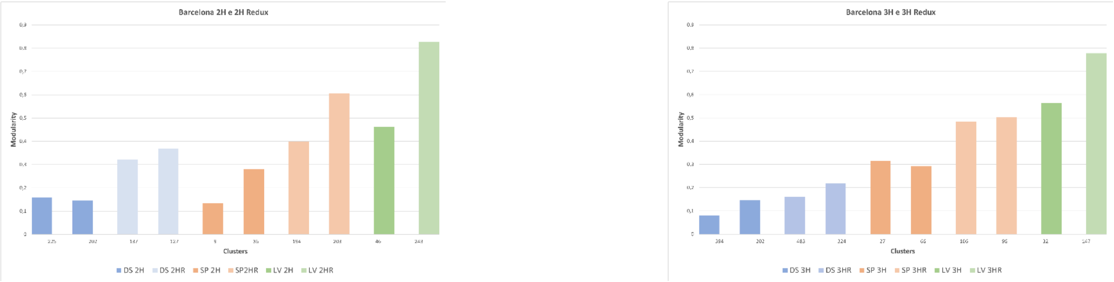
For each experiment, we chose as the *cutoff* parameter for Dominant Set *1.0e-4, 1.0e-8, 1.0e-16* and in some cases even *1.0e-12 and 1.0e-30*, with fixed *epsilon 1.0e-6*. For Spectral clustering we have used the top 3 optimized number of clusters, and the same numbers obtained were used for K-Means. Finally DBSCAN after some testings, a range between *0.3 to 3* has been utilized.



(a) Lisbon 2H

(b) Lisbon 3H

Figure 4.1: Modularity Lisbon 2H/3H



(a) Barcelona 2H

(b) Barcelona 3H

Figure 4.2: Modularity Barcelona 2H/3H

In figure 4.1 we can see that both dominant set and spectral clustering are quite behind the community dectection algorithm in terms of modularity. This is especially reflected on both Lisbon and Barcelona 3H. However there are a few explanations about this poor results, which is strictly related to the database and to the metric itself.

The first one is due to the fact that the data is very sparse, and as a consequence Dominant set is not converging properly. We can notice this thanks to the algorithm internal criterion for the quality of the clusters, which are on average between 0.6 up to 0.8. More details about the data results can be seen from table **A.1** to **A.5**. In particular we observed in Dominant set an improvement of the average coherence of almost 0.10 from Barcelona 2H to 3H, while the reduced form of the dataset showed an increase of 0.20 of internal coherence. An interesting data related observation is that the Lisbon datasets behaved in the opposite way. Here, in fact, we observed an improvement of 0.25 in the base one reaching an average internal coherence of 0.84. Instead the reduced dataset went from a 0.54 to an 0.60. This behaviour could be determined from the fact that while removing some relations, we broke the graphs structure.

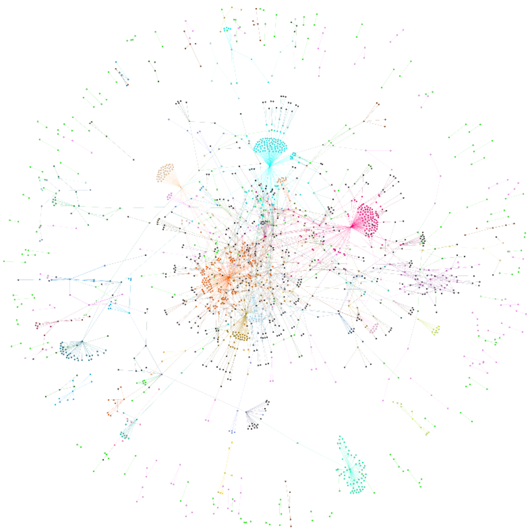


Figure 4.3: DS - cutoff: 1.0e-8

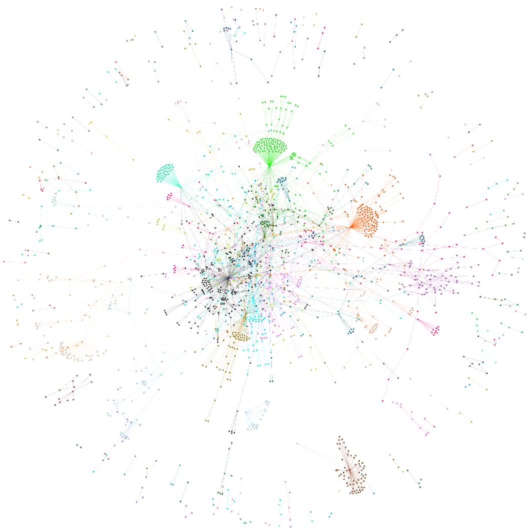


Figure 4.4: Louvain view

Another data dependant problem with the dataset structure, related to both 2H and 3H variants of the datasets, are the relations. When analysing the datasets, we mentioned when commenting tables **4.3** and **4.4**, that some problems could happen when clustering, since many entities of type *Place* are connected to an entity of type *Knowledge*. However most of the time a direct connection between *Place-Place* is not present. As a consequence, this results in one entity with many incoming connections. This can become a limitation somehow, at least in terms of modularity, since Dominant set clustering strategy, mentioned in **2.4.5**, will find first many small strongly connected clusters and finally the lasts clusters will be the ones discarded. The problem with modularity metric instead is due to the fact that its affected negatively by the fragmentation of the data. However we expected Dominant set to do exactly this. In figure 4.3 and 4.4 we can observe

that the partition found by our approach is not too dissimilar to the network analysis algorithm. Although there is clearly some misclustering in the outside data, overall the algorithm is working as expected. However the chosen metric for this unsupervised task is not optimal for the algorithm itself, since their view on what is a cluster diverges.
Comparing Dominant set with Spectral clustering and Louvain Community, the values obtained put into evidence that our strategy is surely limited by the data and by the available metric. In contrast to our partition of many micro-communities, the network algorithm clearly fits the best results and shows that Louvain is more fitted for the task of find macro-communities on the KG based on the structure/-topology of the KG.

The choice of the 2H or 3H variant seems to have significant effects on the results too. In dominant set having a bigger and sparse dataset seems to have a marginal negative effect, at least in terms of modularity for the limitations just mentioned. In general the 2H structure performs the best in all of its test regarding modularity. However the algorithm clearly is more "confident" on the clusters generated in the 3H variants. Spectral clustering showed small improvements going from 2H to 3H in the base dataset, while the reduced one got better scores in the 2H form. Finally Louvain community in the base dataset always showed an improvement going to 2H to 3H. Instead the reduced dataset, while getting better scores, showed no significant difference between the 2H and 3H form.

Overall one last observation is that in both spectral clustering and louvain, going from the normal dataset to its reduced form, increased the number of clusters found by the algorithms. This clearly shows that the the removal of the relations results in the fragmentation of the clusters.

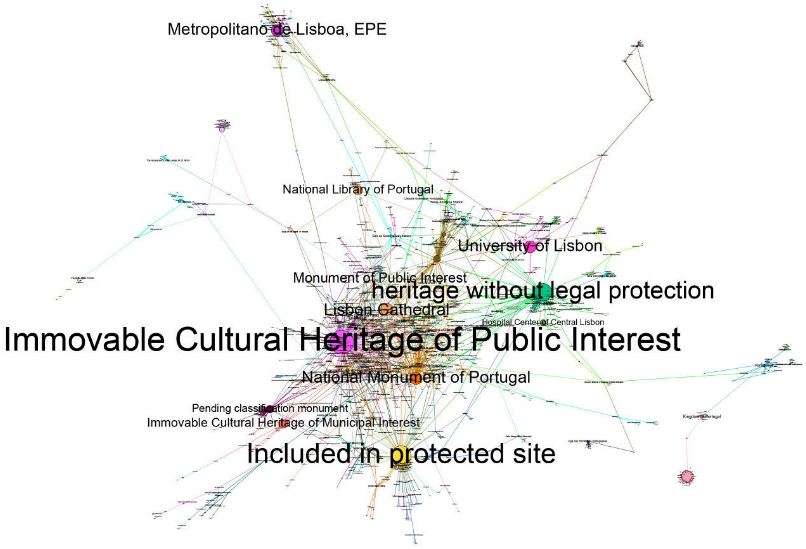
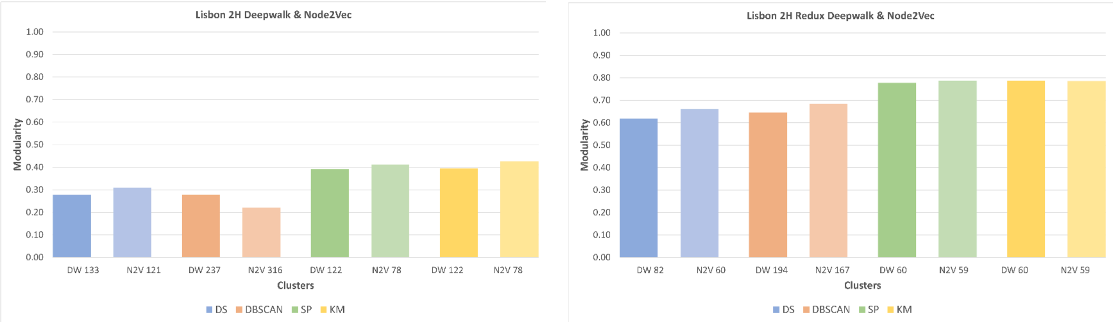

Metropolitano de Lisboa, EPE
National Library of Portugal
University of Lisbon
Monument of Public Interest
heritage without legal protection
Lisbon Cathedral
Hospital Center of Central Lisbon
Immovable Cultural Heritage of Public Interest
National Monument of Portugal
Pending classification monument
Immovable Cultural Heritage of Municipal Interest
Included in protected site
Kingdom of Portugal

Figure 4.5: Macro communities found by DS on the graph structure

4.2.2 Topology Embeddings

After the preliminary step of clustering directly from the graph structure as it is, we can now continue with the concept of having the similarity matrix of pairwise similarities, as discussed in the section 3.1.2. For the embedding techniques we introduced K-Means and DBSCAN, although Louvain couldn't be utilized it since no available method was present at the time of testing.
We exploited the Node2Vec [18] and DeepWalk [38] algorithms already configured with their best hyper parameters as described in their respective papers.

Figure 4.6 shows that DeepWalk is able to retain the graph structure very similarly to Node2Vec, which results in better clusters in term of modularity. This is interesting being one of the earliest techniques in the state of the art for graph embedding generation.



(a) Lisbon 2H

(b) Lisbon 2H Reduced

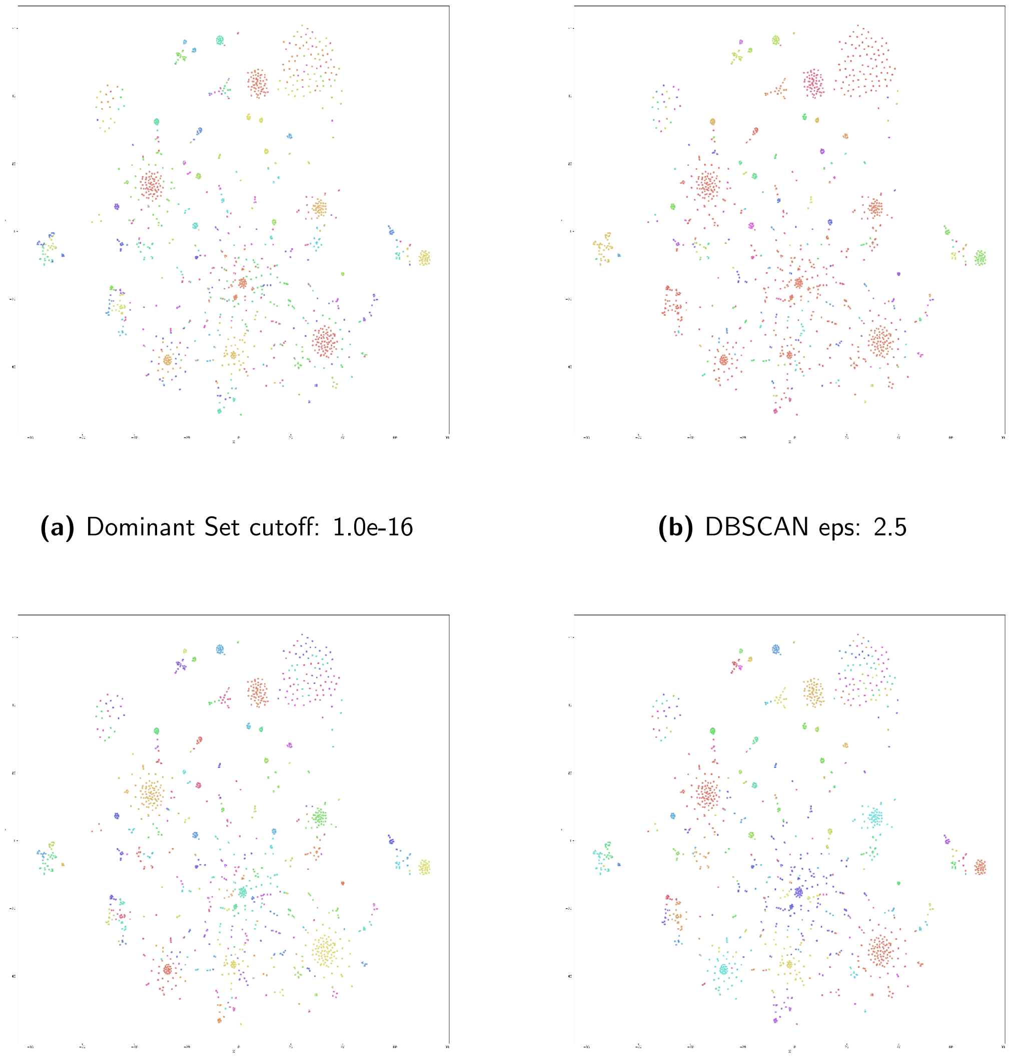
Figure 4.6: Modularity of topology embeddings

Taking into consideration the modularity metric, it shows that Spectral clustering and K-Means had the best results followed by DBSCAN and Dominant Set. One can notice that even though the number of clusters changed, modularity stayed mostly the same, specially in the reduced dataset. This is probably due to the algorithms finding different small partitions in each different run. Putting DBSCAN in comparison to Dominant set, we expected them to have similar results. However one can notice that Dominant set remain much more stable, even changing parameters, while DBSCAN had very inconsistent results.

Considering the other quality metrics is important to remind that Silhouette, Davies Bouldin and Calinski-Harabasz have the tendency to work better a higher and with convex clusters, therefore with a larger occurrence of singletons we will still obtain optimal results for these metrics. Overall Silhouette stays around 0.30, reaching a maximum value of 0.37 with Dominant set when the partitioning is too high, as one can see in table A.6.

Figure 4.7 shows Dominant set, followed by DBSCAN, Spectral clustering and K-Means. While Spectral and K-Means are almost identical, we can notice that

Dominant set is partitioning the data a bit more. DBSCAN in contrast, as expected, is agglomerating many groups together.
In general some of these partitions clearly have sense, like the ones on the top left in figure and on the bottom-left, while others, like the big cluster on the top right doesn't looks right. Unfortunately the TSNE representation of the data in the 2D space doesn't capture how distant the data is between each other in the embedding space.



(a) Dominant Set cutoff: 1.0e-16

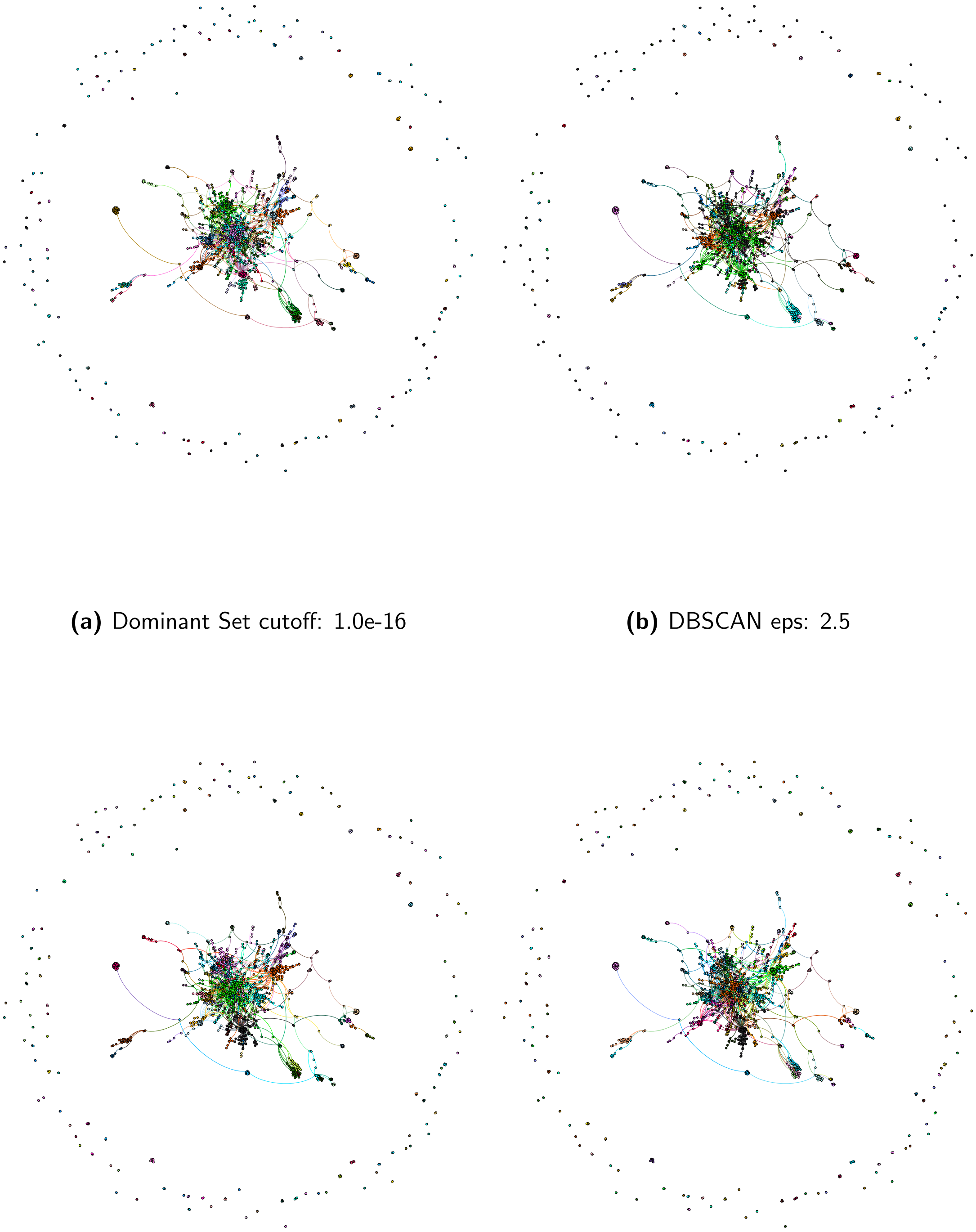
(b) DBSCAN eps: 2.5

(c) Spectral Clustering k: 122

(d) K-Means k: 122

Figure 4.7: DeepWalk Topology embedding clustering

On a practical demonstration we have that the partitioning of the graph with the help of topology embeddings translate quite well into the application itself. Mapping the clusters to the graph, shows, as seen in figure 4.8 that Dominant set is clearly able to find the main communities in the center of the graph, while is misclustering some of the small data on the sides.



(a) Dominant Set cutoff: 1.0e-16

(b) DBSCAN eps: 2.5

(c) K-Means k: 122

(d) Spectral Clustering k: 122

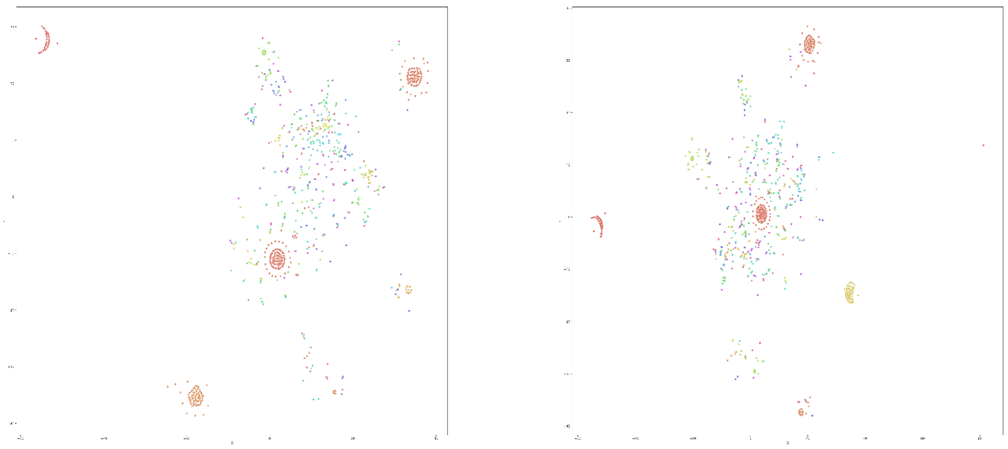
Figure 4.8: DeepWalk Topology embedding clustering mapped on the graph

A different behaviour happens when the number of partitions found is higher. In those situations it is able to cluster the small data, while clustering a bit worse the center of the graph. Obviously this is a trade off that has to be made in order to improve the first or the later.

In the reduced datasets (**A.6**) we observed that here Dominant set performed quite fair reaching up to 0.61 on Lisbon 2H.
Performance wise, Spectral clustering and K-Means performed on average 0.15 more for both the normal and reduced datasets in respect to Dominant set and DBSCAN. In the other quality metrics, Dominant set is on average almost better than DBSCAN but worse than the other two algorithms (which are running with the optimized number of clusters). Some bias can be notice in the Silhouette score by the considerations raised before. Still though, comparing the communities obtained with the ones found by Louvain community in the structure test, we can clearly see that the topology embedding is working well for these datasets and the clustering algorithms are still able define communities with qualitative meaning. However on the negative side, the partition of the communities is higher in Dominant set and DBSCAN.

4.2.3 Semantics Embeddings

At this point we can introduce the words embeddings that we described early in section **2.2.2**. We performed clustering based on the semantic meaning of the description attribute in KG nodes, where the KG is transformed to a graph as written in section 3.1.2. We exploited Word2Vec [28] [29], Google's pre-trained word embeddings and FastText [7] pre-trained as well, using Python's Gensim library. Then we apply the different algorithms for clustering on top of the description embeddings.



(a) FastText

(b) Word2Vec

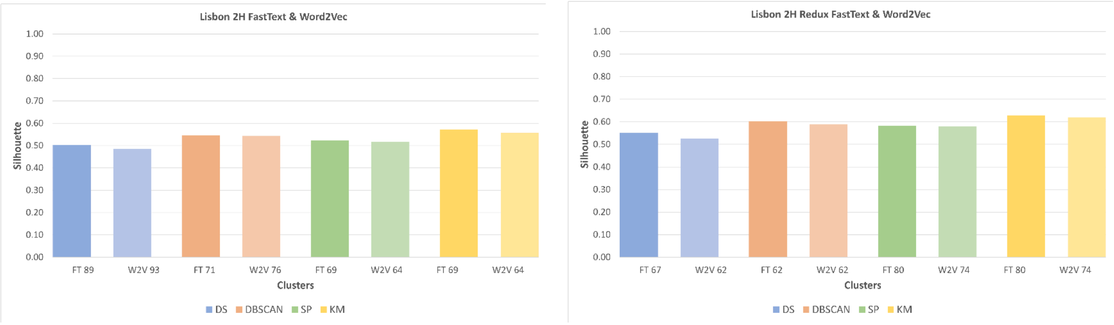
Figure 4.9: T-SNE representation of Word Embeddings, mapped with Dominant Set clusters

Modularity here doesn't get good scores, since we noticed that there's no a real correlation between elements on the graph and their description. For example we can have two nodes connected between each other like Antonio Gaudì and Park Güell. We obviously know for a fact that Gaudì is the architect of the famous Park Güell in Barcelona, however their respective descriptions are:

- **Antonio Gaudí:** Spanish architect;

- **Park Güell:** public park system in Barcelona, Spain.

Here Dominant set (A.14) performed very similarly for both the types of datasets and the embedding type. Overall we can say that Word2Vec has by very little some slightly better results based on the Silhouette score. Figure 4.9 shows that the behaviour of the data is indeed similar in both situations. Figure 4.10 shows how all the algorithms are able to separate the data based on the silhouette score.



(a) Lisbon 2H

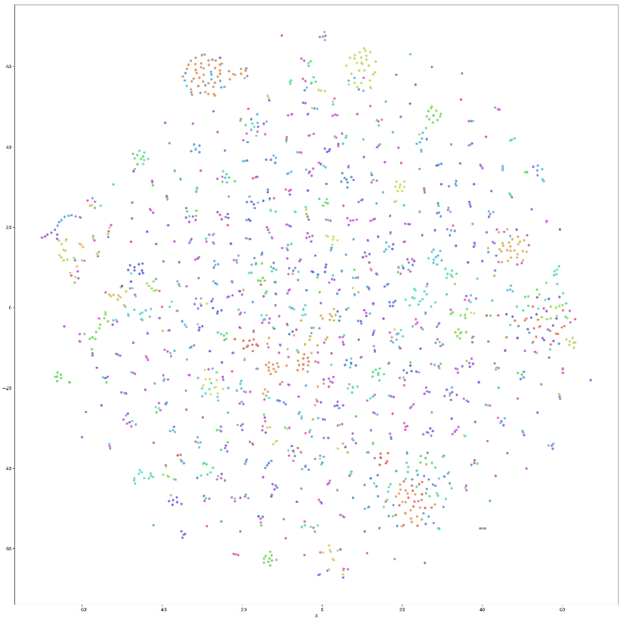
(b) Lisbon 2H Reduced

Figure 4.10: Silhouette of word embeddings

Overall, considering that the descriptions are still very much ”naive”, this early step shows that grouping the KG related entities together is definitely possible, and with some improvement on the quality of the descriptions, the precision could improve extensively.

4.2.4 Translational Embeddings

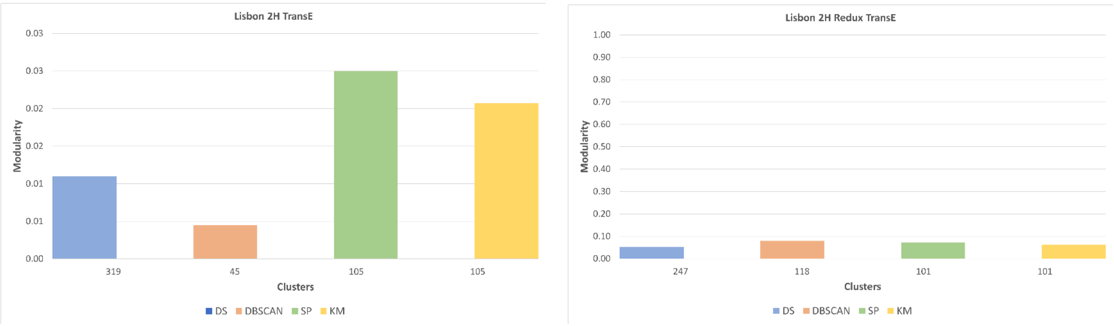
To explore graph embedding techniques we reviewed in section **2.2.3** we have the translational models that model graph relationships by interpreting them as translations in the embedding space. Once again the KG has been transformed into a simple undirected graph, starting from the embeddings (3.1.2). In the MEMEX project we have applied and evaluated all of them on the link prediction task [42], but for summarize this thesis work, only transE will we considered, since is the best performing method based on our KG. Consequently, we run once again all the described clustering algorithms on top of the pre-trained embeddings from TransE. Clustering in this regard shows very little correspondence with the modularity of the graph for all the algorithms. Some improvements can be seen in the reduced dataset, where we achieved a maximum of 0.07 and 0.073 with dominant set and spectral clustering respectively. Not even the other quality metrics (A.22) seem to have a positive score in this regard.



(a) Dominant Set

Figure 4.11: TransE embeddings clustering

In general we noticed that the translational embeddings get a lot of overlapping clusters. When analysing the problem we first though this was mainly happening in base of the observations made in 4.3, since having many relations that connect everything in the graph can reduce performance. We also thought that a critical factor is due the size of the dataset, which is still too small, and the training is not sufficiently enough to learn qualitatively embeddings. Although all this observations can surely contribute to the general limitations of the problem, we we also notice that, looking at entities of the clusters on the tSNE plot in figure 4.11, most of the time coherent data was in the same region of the plot, but misclassified.



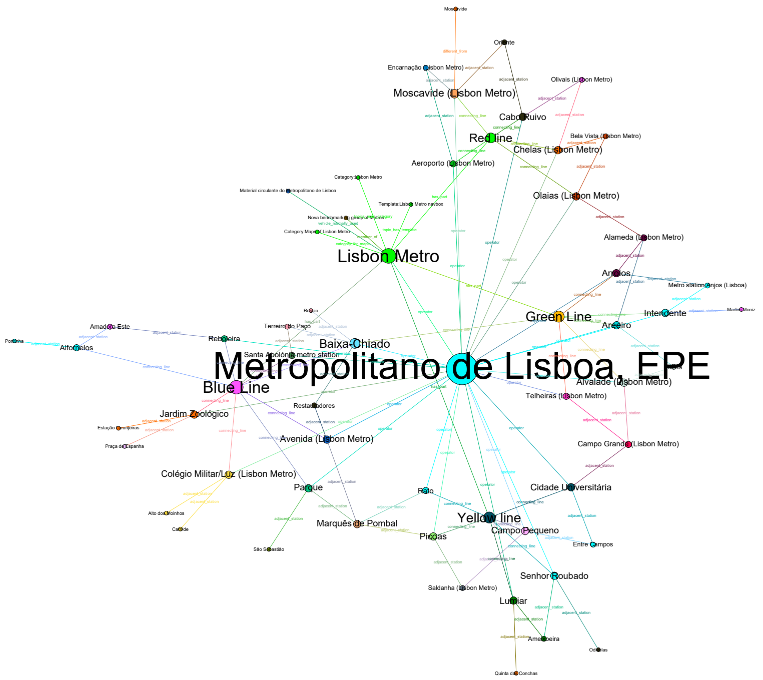
(a) Lisbon 2H

(b) Lisbon 2H Reduced

Figure 4.12: Silhouette of translating embeddings

As a practical example, if we look up closely to the macro community of the "Metropolitan of Lisbon" in figure 4.13, we can see that the partitioning its quite high, though very context specific. As we said, most of this entities were very close in the embedding region. If we look at the problem from another point of view, this could actually be seen as clustering very context specific information. Reasoning about what is happening, we could say that entities that share the similar relations can be found in the same "x axis", but the're being positioned far from each other if some relations are different or are missing. This explains why, even though close in the 2D space, they are not actually clustered together.

For this very specific task, translational embeddings might not be useful by themselves, however since data is actually located closely, this might improve some of the results when considering embedding concatenation.



(a) Dominant Set cutoff: 1.0e-8

Figure 4.13: TransE embeddings clustering mapped to the graph structure

4.2.5 Embedding combinations

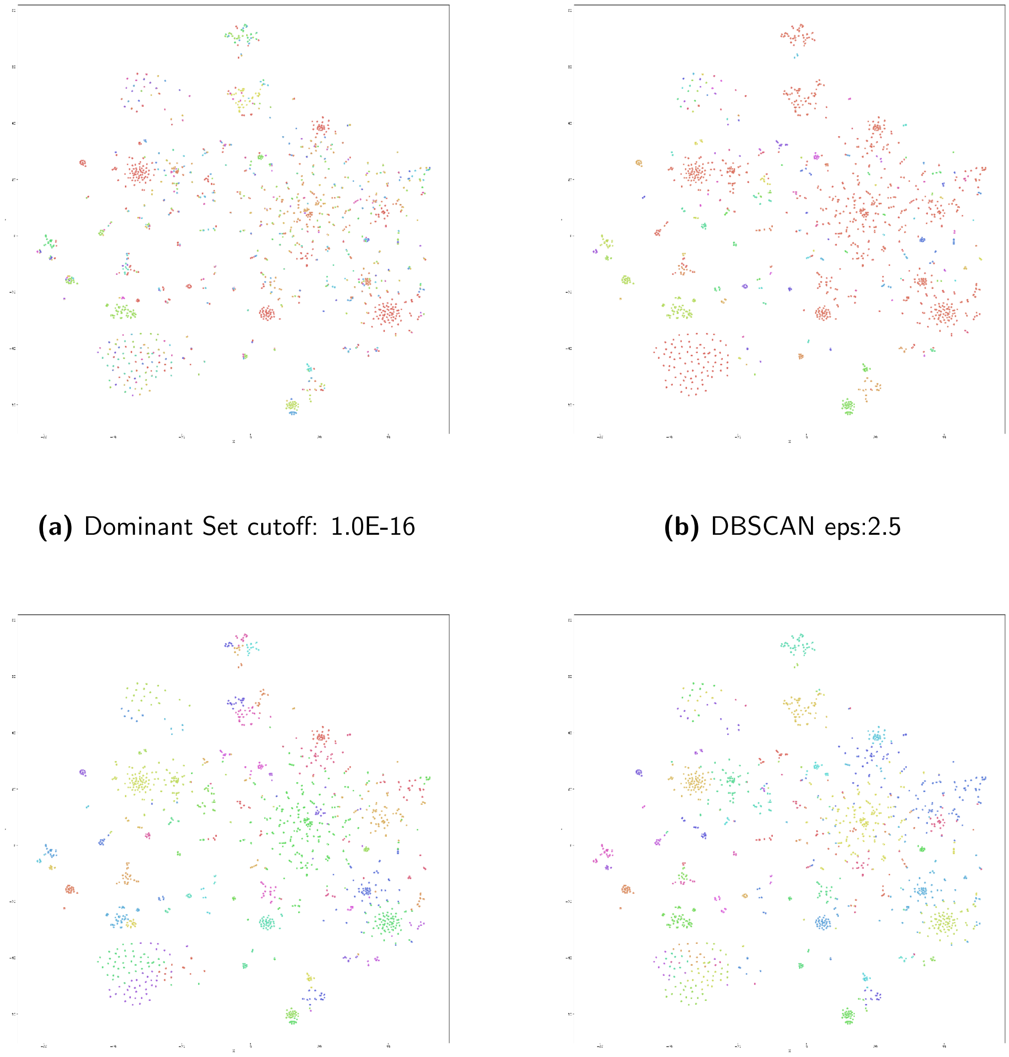
In section 3.1.3 we introduced the idea of transforming a KG into a graph using the embeddings combination. In this section we will discuss about the results we got after the combination of the embeddings, to try to investigate if some useful data could be obtained. We expect, that if the embeddings are correlated, their concatenation will move the data in the embedding space, facilitating the discovery of new communities and clustering of the data.

Through the combination of the topology embeddings and the description embeddings, we noticed in figure 4.14 that the modularity obtained was worse than the topology ones A.6, but overall highly better than word embeddings A.14. However, while Spectral clustering and K-Means showed basically identical results, DBSCAN performance was significantly worse. Finally here Dominant set is not able to cluster the data efficiently. Here probably a limitation on clustering on top of the similarity matrix comes into factor, since its not able to cluster poorly separated data. Still the performance is not substantially different from DBSCAN if we look into the Silhouette score. So its probable that due to the nature of DBSCAN, some of the data clustered was lucky enough to be also connected. Most of this behaviors can be noticed in from table A.26 to table A.41. Performance wise its not that satisfactory, but since we were expecting it, yet some coherent data has been retrieved. The usage of DeepWalk or Node2Vec embeddings did not translate into a significant difference, although the best results were achieved by the combination of DeepWalk with FastText. Considering the silhouette, the metric is more or less between -0.1 to 0.1 for both DBSCAN and Dominant Set and between 0.1 to 0.2 for K-Means and Spectral Clustering. The last two put into evidence that some clusters could be retrieved according to the metric.
This is then confirmed by the results on the tsne representation of clustering on the embeddings, that can be seen in figure 4.14.

Although the embeddings of the data are still a bit noisy and sparse, they clearly show that the combination itself has some form of correlation. It is also an improvement, especially if we confront if with the previous description embeddings test.

This is important, since it proves that with improved descriptions and tags, a clear clustering problem could be definitely solved.

It also puts into evidence our the current limitations of our the description embeddings like we clarified before.



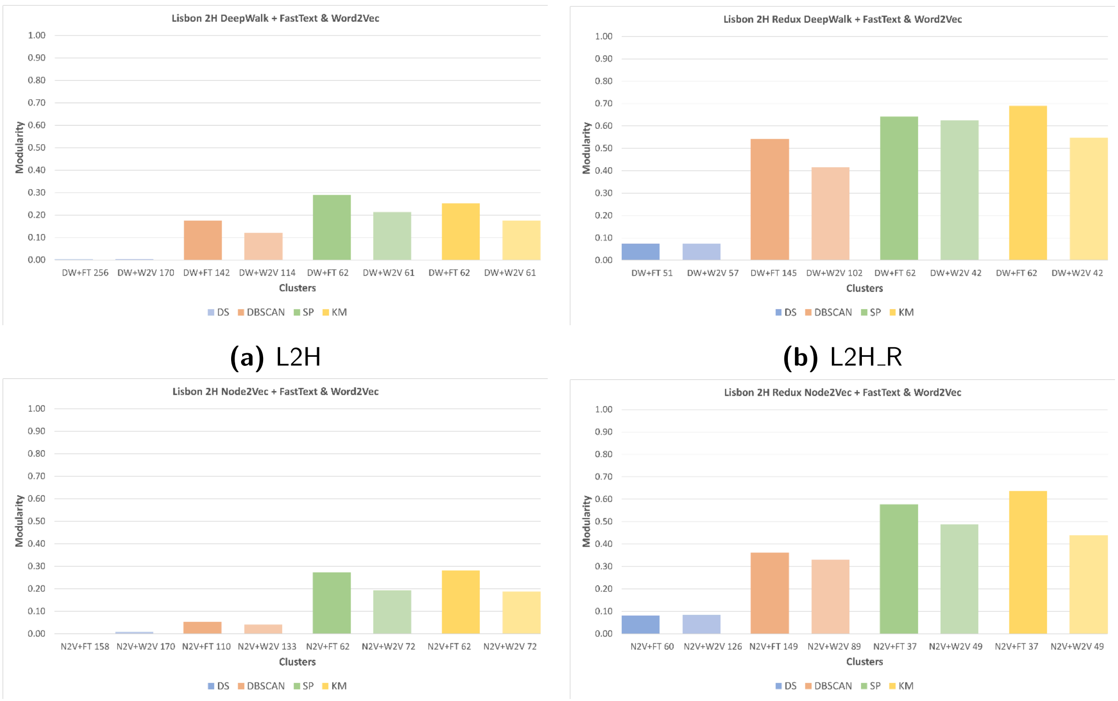
(a) Dominant Set cutoff: 1.0E-16

(b) DBSCAN eps:2.5

(c) Spectral Clustering k: 62

(d) K-Means k: 62

Figure 4.14: Clustering on DeepWalk and FastText embeddings



(a) L2H

(b) L2H_R

(c) L2H

(d) L2H_R

Figure 4.15: Modularity of clustering topology embeddings with word embeddings

The second combination that we have performed is between the topology's embeddings and the translational embeddings (A.42) and between the translational embeddings and the word embeddings descriptions (A.50).

Considering textual descriptions with transE embeddings, on a first look on the tables results **A.50** it looks like the scores are barely significant. However looking at the data separation on figure 4.16 it is evident that some of the data has moved in the space, making possible the discovery of new partitions. As a matter of fact here silhouette has scored 0.175 with Dominant set, with only DBSCAN getting close with a 0.146. Still the data itself is not good, but on some basic level the combination has some possible positive effects.

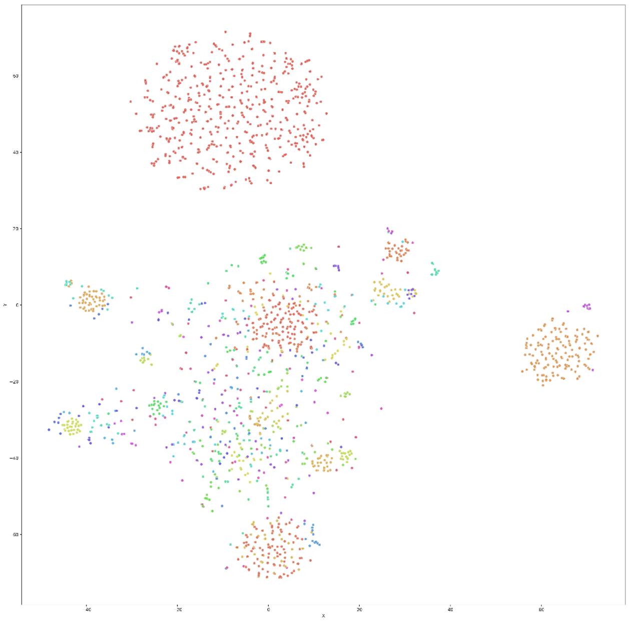
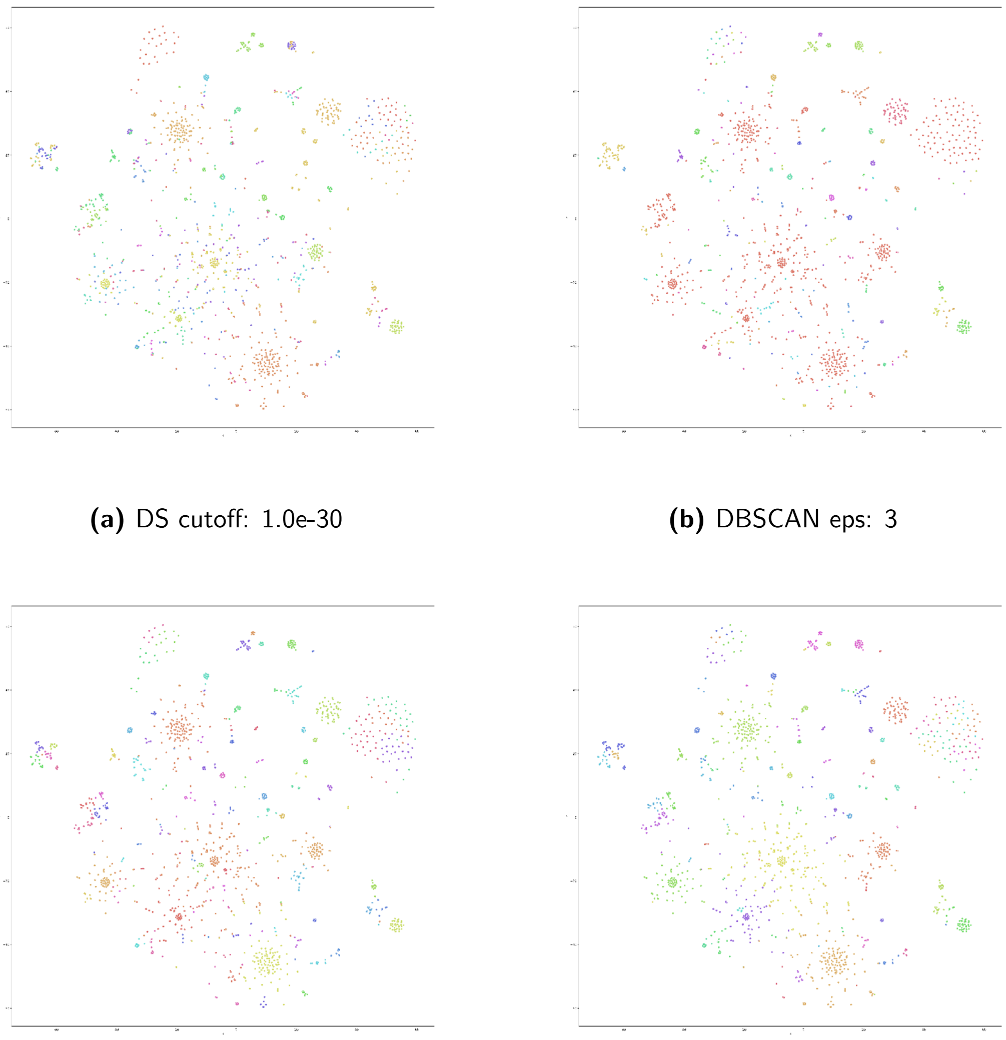


Figure 4.16: DS cutoff: 1.0e-16 - Clustering on FastText + TransE embeddings

The combination of the topology with the translational embeddings instead shows some very interesting results (**A.42** to **A.49**). In particular, although all scores are a bit behind the topology ones in terms of modularity, we can clearly see that the combination of this two types of embeddings, proves that there's indeed correlation between them. This also reflects that the observations that we made in 4.2.4 are not completely wrong.

Here Dominant set performance is pretty much on the same level of the seen before in terms of modularity. Silhouette, Davies and CHI scores instead decreased marginally. Here Spectral clustering performed the same in the combination of DeepWalk with TransE and is not drastically different in the Node2Vec - TransE one.

Generally speaking the combination of translational embeddings with the topology embeddings for KG could represent a good solution for the clustering problem, since, as we can see in figure 4.17, the embeddings look very promising. In fact, having also the information about the relations of the graph through the translating embeddings, seems to be helping the data separation problem on the topology embeddings. In terms of improving modularity it might be scoring a little worse, however if look at the data, the clusters are definitely separated in a better way and much distant from each other.



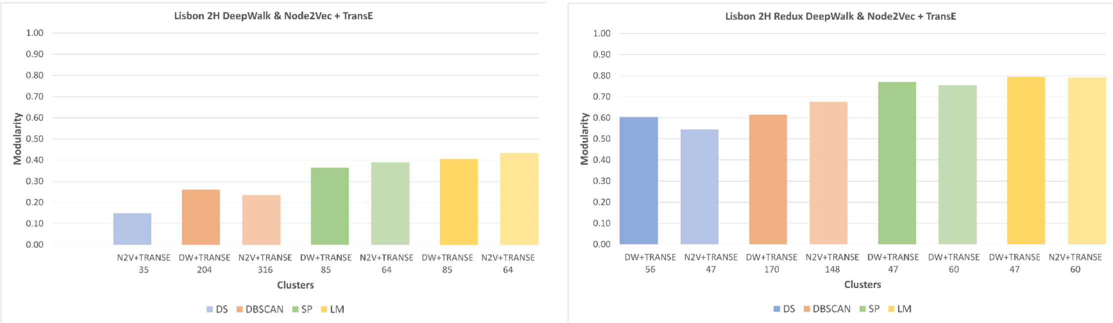
(a) DS cutoff: 1.0e-30

(b) DBSCAN eps: 3

(c) Spectral Clustering k: 77

(d) K-Means k: 77

Figure 4.17: Clustering on DeepWalk + TransE embeddings



(a) L2H

(b) L2H Reduced

Figure 4.18: Modularity of clustering transE embeddings combined with topology embeddings

The final step of this experimental setup shows the combination of all the three types of embeddings. With this experiment we wanted to check if all the information retained in the KGs could be useful if merged together. So far we have seen how the combination of two embeddings have moved the data in a significant way in some aspects, especially the just seen one with topology and translational embeddings. The combination of the Knowledge Graph with the semantics about CH and the link prediction embeddings still referred to CH is indeed ambitious and probably difficult to achieve a good data separation.

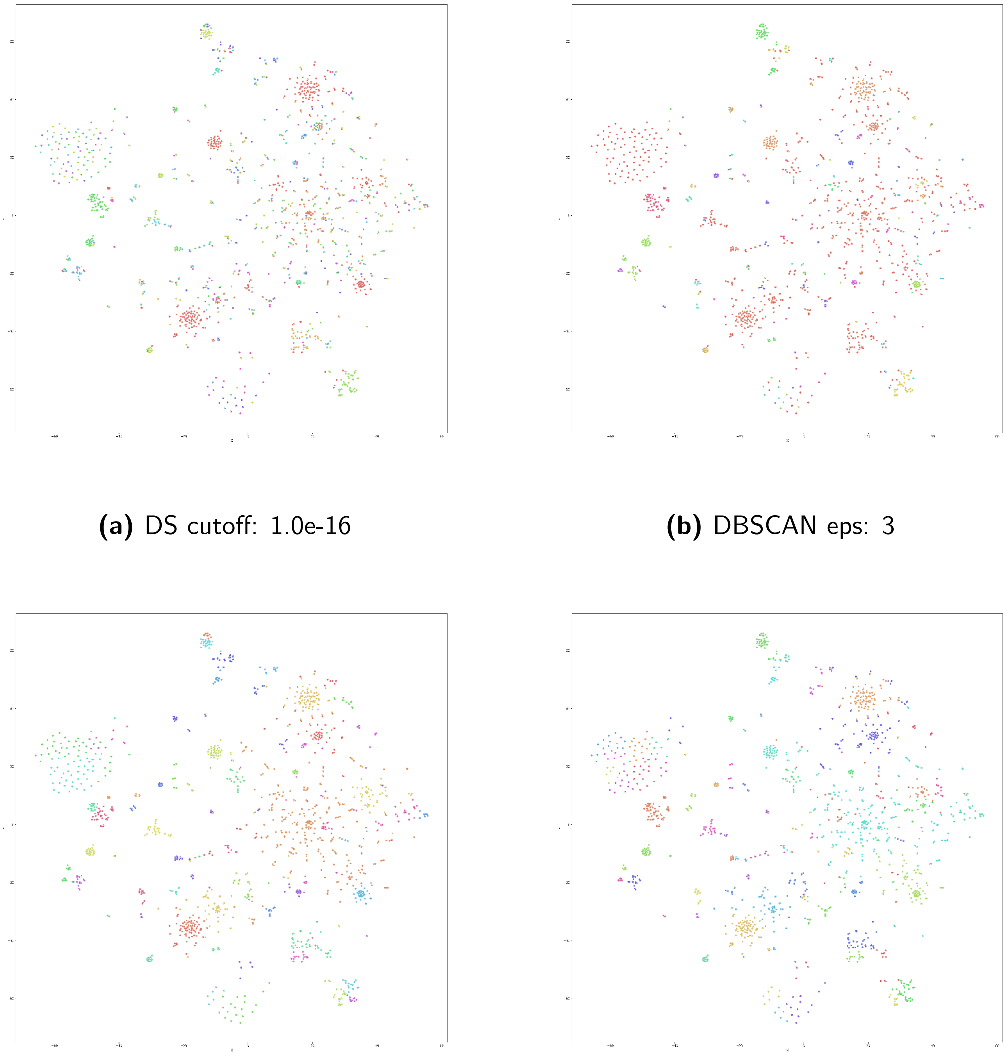
Results are indeed difficult to understand, since the quality metrics themselves are not always too useful. Overall we definitely can say that modularity is lost once again for Dominant set, since we can observe the same behaviour presented in table A.26. The same can be said about silhouette, but the embeddings themselves can be considered on the same level of the one obtained with the combination of the topology with the description embeddings, or the traslantional embeddings with the description.

In fact, some possible communities can clearly be seen in figure **4.19**. One observation that can be said is that the quality of the embeddings is worse, since it is more fragmented in certain areas, especially if we compared it to the ones obtained with the combination of topology with translational embeddings, which where totally some unexpected results.

Definitely being the task not trivial and usually not performed in Knowledge Graphs we have seen, that the clustering problem itself, can be formulated in various combinations, depending on what one is looking for in the data. Fortunately KG are a powerful data structure that if well contextualized and created, could retrieve very useful information.

Dominant Set clearly has proven to be working quite well and even though sometimes the pealing off strategy leads to the partition of the clusters, in general we have observed that very useful information can be obtained by this method.

Finally further investigations should be done, due to the nature of the KG. Improving the descriptions of the entities could show some good results. Being the nodes connected in the graph, more ad hoc information about them might improve the topology clustering and the modularity. Also, thanks to the observations that we noticed about the translational embeddings and their performance when concatenated with the topology embeddings, we can say that clustering on a graph with extra added information not only by its structure, but also about its relations, could become very useful when looking at the task of finding communities and improving the networks modularity.
In terms of the MEMEX project, this research activity clearly showed some interesting ways in which the KG can be used.



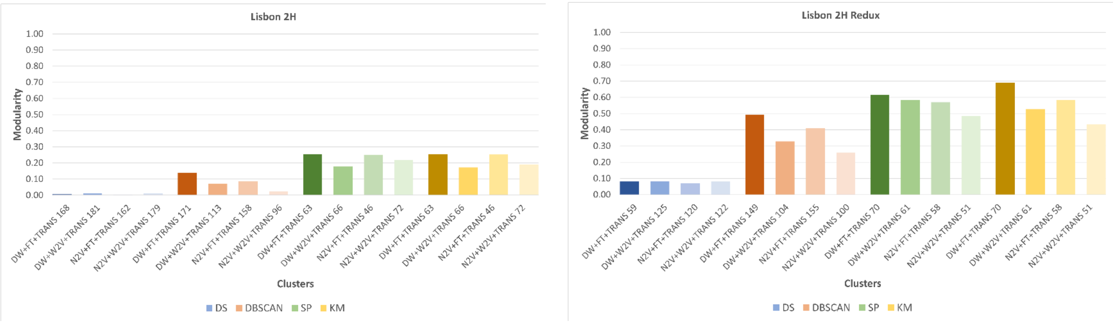
(a) DS cutoff: 1.0e-16

(b) DBSCAN eps: 3

(c) Spectral Clustering k: 63

(d) KMEANS k: 63

Figure 4.19: Clustering on DeepWalk + FastText + TransE embeddings



(a) L2H

(b) L2H_R

Figure 4.20: Modularity of clustering topology combined with word embeddings and transE

4.3 Result Analysis graph querying

Since the aim of the project is to allow the creation of a narrative around CH places or objects, we utilised local clustering techniques as seen in section 2.5, in particular Dominant set (section 3.1.4) in comparison with Personalized PageRank [5] for extracting an informative subgraph from the whole KG including the query/seed nodes provided by the user (which representing CH places or objects). The subgraph will allow us to explain the relation between the query nodes. In addition, clustering techniques based on the graph topology and node attributes can be further applied on top of the subgraph to group the strongly connected or the semantically similar nodes together in order to facilitate the story creation process.

In this section the works that has been done is still limited to a toy dataset where we were trying to analyse the concept of querying the graph in a compact way. We chose for the query the node entity *"Park Güell"* and *"University of Barcelona"*. As shown in figure 4.23, the subgraph extracted using Personalized PageRank explains the relatedness between two query nodes.

Dominant set graph extraction in comparison to Personalized PageRank is a little more conservative, and fits more the concept of clique, as intended by the algorithm. In fact with a strict choice of the parameters, we can see two clusters: *"Park Güell"* connected to *Works of Antonio Guadì* and the artist itself, and the *University of Barcelona* connected to *Josep Maria Jujol.* This can be revisited as: *"Park Güell" is a work of "Antoni Gaudi", who also contributed in "Casa Batlló" with "Josep Maria Jujol", who was an employer in the "University of Barcelona".* Instead if we are more loose on the parameters, Dominant set is able to retrieve much more information, although not necessarily relevant to the story creation/enriching.

In both cases the approach seems very promising, since then, if combined with the clustering work, very different kind of data could be retrieved. Comparing Dominant set with PageRank we can see that both available methods for local clustering are quite interesting. PageRank allowed us to extract a bigger subgraph that includes user CH query nodes, while Dominant set allowed to restrict the information even further.

Moreover the subgraph can be further clustered based on its topology and/or node semantics as we experimented before. The Dominant Set clustering technique can found successfully group similar nodes together based on their location on the subgraph or their semantic similarity. Each cluster in the subgraph can be used to write a sentence or a part of the story.

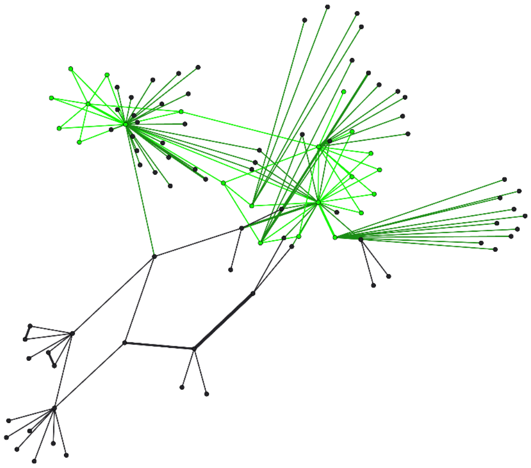


Figure 4.21: Local Clustering

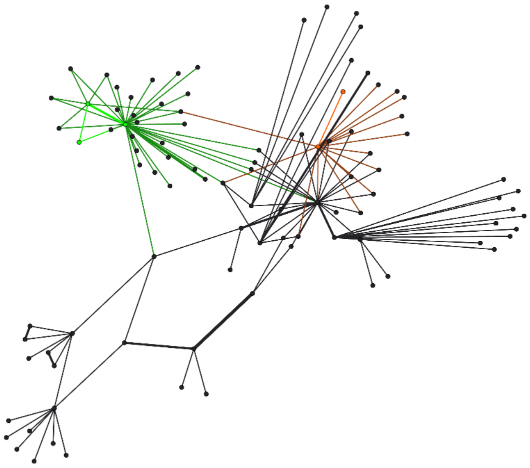


Figure 4.22: Constrained Dominant Set

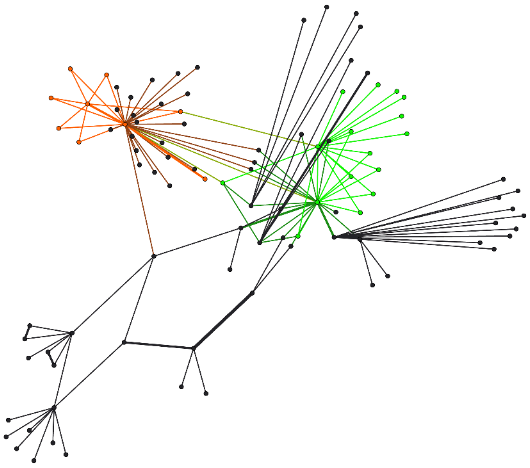


Figure 4.23: Constrained Dominant Set

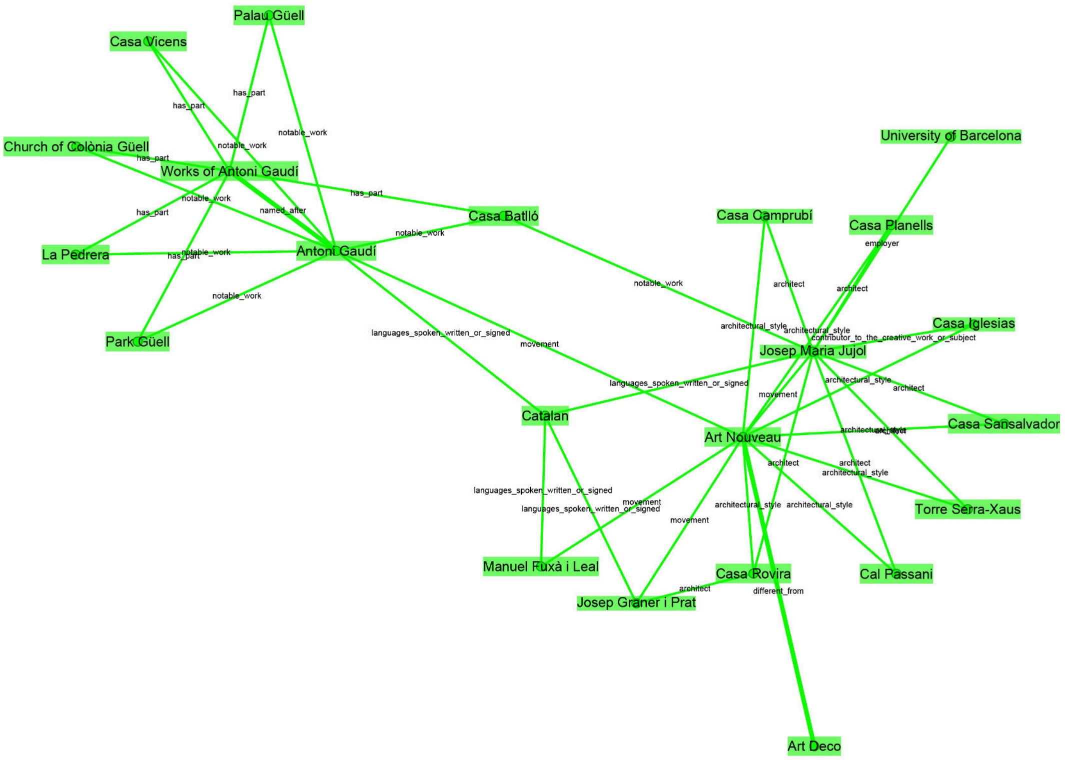


Figure 4.24: Local Clustering detailed view

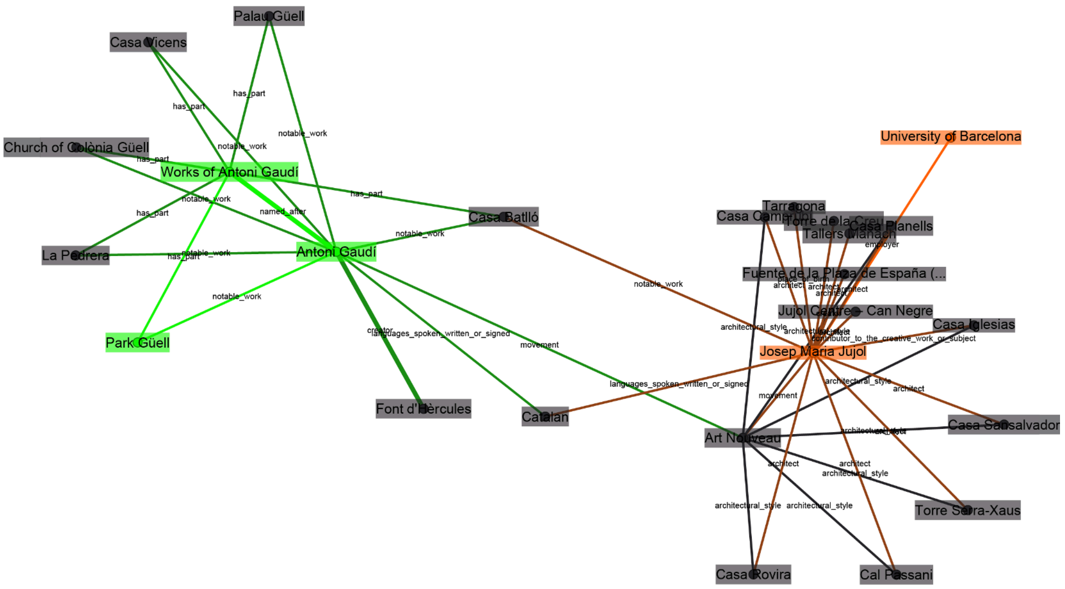


Figure 4.25: Constrained Dominant Set Detailed view

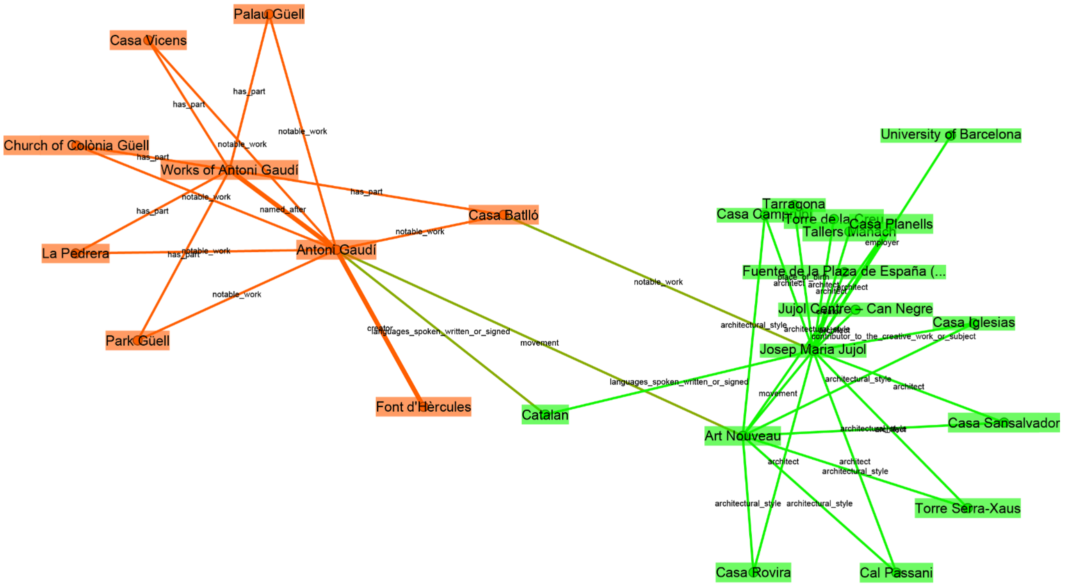


Figure 4.26: Constrained Dominant Set Detailed view

Chapter 5

Conclusions

In this thesis, we explored the possibility of extending the Dominant Set algorithm for the Knowledge Graphs, with the strategies presented in chapter 2 used to build the pairwise similarity matrices needed to feed our clustering algorithm. We did that by taking advantage of the various representations that a Knowledge Graph has, therefore adapting them with the features learning techniques that suited best the view of the KG.

We first explored how such strategies behave in our idealized scenarios with controlled data to then test them in a production-like environment. By doing so, we reviewed some of the *state-of-the-art* for the representational learning on graphs, word embeddings used for the specific context of cultural heritage and then translating embeddings for modelling multi-relational data.
We then discussed how we implemented Dominant Set in the context of KG. The algorithm has never been applied before in this research area and, although it was not a trivial task, it clearly shows that with more research and refinements, good clustering can be achieved.

At this point we explored the idea of concatenating the various types of embeddings, in order to verify if, being by definition a KG a collection of interlinked data that can be descriptions, entities and relations, their combination could somehow extend the concept of modularity of the network itself. Therefore, to stay in line as well with the project MEMEX goals, we asked ourself how to retrieved efficiently the entities needed to enrich the stories in the story-telling process. As a direct consequence we began exploring the concept of querying a graph.

Ultimately we summarized our findings in chapter 4, arriving at the conclusion that finding communities directly from each of the representations of a KG is a viable option. Moreover in order to improve modularity, we have seen that each of the embeddings combinations had it owns trade-offs.

The overall conclusion that we observed from this work is that it is possible to retrieve different kinds of information from each of the base techniques, in the CH

context. Specifically in order to improve modularity we observed that, although with some limitations, the embedding combination indeed retained the topology informations. In particular the translating embeddings combined with the representational learning ones, helped to moved the data in the embedding space in a way that facilitated the capture of clusters by the techniques we used. Furthermore, it seems plausible to be being able to improve the modularity of the graph itself taking into consideration the various types of relationships of the graph.
Also, although results were less than satisfactory, an open path remains the topic of modularity of the graph in regards to the descriptions of the entities each node contains.

Finally for graph clustering, we have that the combination of all the types of embeddings, still showed us some coherent clusters according to the metric, although the quality of the embeddings was less enchanting.

On the other side, considering local clustering, one last important consideration is that all of the clustering on the graph could be effectively replicated into a more constrained environment thanks to the utilization of these types of techniques.

Here we have seen that Dominant set is perfectly able to retrieve a subgraph, or more than one, with a low computational cost that ranges from one to only a couple iterations of the algorithm. This improves the performance drastically when trying to retrieve information quickly. Furthermore, if compared to the Personalized Page Rank (PPR) technique that we have used, we have noticed how our approach gives back more qualitative clusters, with definitely less choice, but more coherent ones that could help the story creation/enriching process.

For future developments, we would like to investigate further more this approaches. Specifically we believe that improving the quality of the descriptions into a more CH oriented model could improve the concept of modularity since like we have seen, the data connected is generally also semantically connected. Also much more work can still be done with the translational embeddings and the topology ones, which proved successfully to cluster communities thanks to the information of the topology and the information about the relationships.

Finally for the MEMEX project the focus has moved more into the local clustering approach. This last one shares in common many of the requirements of the project, making it a very promising approach, not only in terms of information retrieved, but also in terms of performance. Furthermore the research then could be even more precise introducing the localization coordinates, that we already successfully tried with just the descriptions similarity.

While continuing improving our KG, we will start looking into how to integrate better our algorithm into the concept of KGs. Specifically we will start looking into the multimodal KGs, adding the information of images, audio and videos on top of the already present structure. This will allow us to focus on answers to our research strategy results and observe how Dominant set behaves in comparison to other techniques, for the final goal of improving the story generation process.

Bibliography

[1] Embeddings, 2021. URL https://developers.google.com/machine-learning/crash-course/embeddings/video-lecture.

[2] Memex project, 2021. URL https://memexproject.eu/en/home.

[3] Färber M. Bartscherer F Menne C. Rettinger A. Linked data quality of dbpedia, freebase, opencyc, wikidata, and yago. *Semantic Web Journal*, page pp. 77–129, 2018. doi: https://doi.org/10.3233/SW-170275.

[4] Richard D. Alba. A graph-theoretic definition of a sociometric clique. *Journal of Mathematical Sociology*, 3:3–113, 1973.

[5] R. Andersen, F. Chung, and K. Lang. Local graph partitioning using pagerank vectors. In *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*, pages 475–486, 2006. doi: 10.1109/FOCS.2006.44.

[6] Jyotirmoy Banerjee, Camiel Klink, Edward D. Peters, Wiro J. Niessen, Adriaan Moelker, and Theo van Walsum. Fast and robust 3d ultrasound registration – block and game theoretic matching. *Medical Image Analysis*, 20(1):173–183, 2015. ISSN 1361-8415. doi: https://doi.org/10.1016/j.media.2014.11.004. URL https://www.sciencedirect.com/science/article/pii/S1361841514001613.

[7] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information, 2017.

[8] I.M. Bomze. On standard quadratic problems. *Journal of Global Optimization*, 13 (4): 369–387, 1998.

[9] Murty U.S.R. Bondy, J. A. *Graph Theory*. Springer-Verlag London, 1 edition, 2008. ISBN 978-1-84996-690-0.

[10] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Durán, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, page 2787–2795, Red Hook, NY, USA, 2013. Curran Associates Inc.

[11] T. Caliński and J Harabasz. A dendrite method for cluster analysis. *Communications in Statistics*, 3(1):1–27, 1974. doi: 10.1080/03610927408827101. URL https://www.tandfonline.com/doi/abs/10.1080/03610927408827101.

[12] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*. The MIT Press, 2nd edition, 2001. ISBN 0262032937. URL http://www.amazon.com/Introduction-Algorithms-Thomas-H-Cormen/dp/0262032937%3FSubscriptionId%3D13CT5CVB80YFWJEPWS02%26tag%3Dws%26linkCode%3Dxm2%26camp%3D2025%26creative%3D165953%26creativeASIN%3D0262032937.

[13] D. L. Davies and D. W. Bouldin. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1(2):224–227, 1979. doi: 10.1109/TPAMI.1979.4766909.

[14] Xin Luna Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014*, pages 601–610, 2014. URL `http://www.cs.cmu.edu/~nlao/publication/2014.kdd.pdf`. Evgeniy Gabrilovich Wilko Horn Ni Lao Kevin Murphy Thomas Strohmann Shaohua Sun Wei Zhang Geremy Heitz.

[15] L. Ehrlinger and W. Woß. Towards a definition of knowledge graphs. *SEMANTiCS (Posters, Demos, SuCCESS)*, vol. 48:pp. 1–4, 2016. doi: https://www.semanticscholar.org/paper/Towards-a-Definition-of-Knowledge-Graphs-Ehrlinger-W%C3%B6%C3%9F/b18e4272a7b9fa2e1c970d258ab5ea99ed5e2284.

[16] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proc. of 2nd International Conference on Knowledge Discovery and*, pages 226–231, 1996.

[17] Jane. Fluegel. Chronology. in: Pablo picasso, museum of modern art (exhibition catalog). *William Rubin (ed.)*, 1, 03 1980. doi: ISBN0-87070-519-9.

[18] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks, 2016.

[19] Paulheim H. Knowledge graph refinement: A survey of approaches and evaluation methods. *Semantic Web Journal*, page pp. 489–508, 2017. doi: https://doi.org/10.3233/SW160218.

[20] R. Hamid, A. Johnson, S. Batta, A. Bobick, C. Isbell, and G. Coleman. Detection and explanation of anomalous activities: representing activities as bags of event n-grams. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 1031–1038 vol. 1, 2005. doi: 10.1109/CVPR.2005.127.

[21] J. A. Hartigan and M. A. Wong. A k-means clustering algorithm. *JSTOR: Applied Statistics*, 28(1):100–108, 1979.

[22] Lenar Iskhakov, Bogumil Kaminski, Maksim Mironov, Pawel Pralat, and Liudmila Prokhorenkova. Local clustering coefficient of spatial preferential attachment model, 2019.

[23] Han J. Kamber M. Pei J. *Data Mining: Concepts and Techniques. 3rd edn.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1 edition, 2011. ISBN 9780123814791.

[24] V.D. Blondel J.L. Guillaume R. Lambiotte E. Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, (10), 2008. doi: https://arxiv.org/pdf/0711.0189.pdf.

[25] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, page 2181–2187. AAAI Press, 2015. ISBN 0262511290.

[26] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, USA, 2008. ISBN 0521865719.

[27] Eyasu Mequanint, Leulseged Tesfaye Alemu, and Marcello Pelillo. Dominant sets for "constrained" image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP, 07 2017. doi: 10.1109/TPAMI.2018.2858243.

[28] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space, 2013.

[29] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality, 2013.

[30] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013. URL `https://proceedings.neurips.cc/paper/2013/file/9aa42b31882ec039965f3c4923ce901b-Paper.pdf`.

[31] Lina Teresa Molinas Comet. Clustering knowledge graphs (seminar paper), 03 2019.

[32] T. S. Motzkin and E. G. Straus. Maxima for graphs and a new proof of a theorem of turán. *Canadian Journal of Mathematics*, 17:533–540, 1965. doi: 10.4153/CJM-1965-053-6.

[33] Salvatore Orlando. Clustering, 2021. URL `https://www.dsi.unive.it/~dm/Slides/2_Cluster.pdf`.

[34] Pavan and Pelillo. Dominant sets and hierarchical clustering. In *Proceedings Ninth IEEE International Conference on Computer Vision*, pages 362–369 vol.1, 2003. doi: 10.1109/ICCV.2003.1238367.

[35] Massimiliano Pavan and Marcello Pelillo. Dominant sets and pairwise clustering. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29:167–172, 02 2007. doi: 10.1109/TPAMI.2007.250608.

[36] Marcello Pelillo. Unsupervised clustering, 2021. URL `https://www.dsi.unive.it/~pelillo/Didattica/Artificial%20Intelligence/2018-2019/Clustering.pdf`.

[37] Charles Perez and Rony Germon. Chapter 7 - graph creation and analysis for linking actors: Application to social data. In Robert Layton and Paul A. Watters, editors, *Automating Open Source Intelligence*, pages 103–129. Syngress, Boston, 2016. ISBN 978-0-12-802916-9. doi: https://doi.org/10.1016/B978-0-12-802916-9.00007-5. URL `https://www.sciencedirect.com/science/article/pii/B9780128029169000075`.

[38] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk. *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, Aug 2014. doi: 10.1145/2623330.2623732. URL `http://dx.doi.org/10.1145/2623330.2623732`.

[39] B. Wang Q. Wang, Z. Mao and L. Guo. Knowledge graph embedding: A survey of approaches and applications. *IEEE TKDE*, vol. 29(no. 12):pp. 2724–2743, 2017. doi: https://persagen.com/files/misc/Wang2017Knowledge.pdf.

[40] Leonardo F.R. Ribeiro, Pedro H.P. Saverese, and Daniel R. Figueiredo. struc2vec. *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Aug 2017. doi: 10.1145/3097983.3098061. URL `http://dx.doi.org/10.1145/3097983.3098061`.

[41] Lior Rokach and Oded Maimon. *Clustering Methods*, pages 321–352. Springer US, Boston, MA, 2005. ISBN 978-0-387-25465-4. doi: 10.1007/0-387-25465-X_15. URL https://doi.org/10.1007/0-387-25465-X_15.

[42] Andrea Rossi, Denilson Barbosa, Donatella Firmani, Antonio Matinata, and Paolo Merialdo. Knowledge graph embedding for link prediction. *ACM Transactions on Knowledge Discovery from Data*, 15(2):1–49, Mar 2021. ISSN 1556-472X. doi: 10.1145/3424672. URL http://dx.doi.org/10.1145/3424672.

[43] Samuel Rota Bulò and Marcello Pelillo. Dominant-set clustering: A review. *European Journal of Operational Research*, 262(1):1–13, 2017. ISSN 0377-2217. doi: https://doi.org/10.1016/j.ejor.2017.03.056. URL https://www.sciencedirect.com/science/article/pii/S0377221717302783.

[44] Peter J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1987. ISSN 0377-0427. doi: https://doi.org/10.1016/0377-0427(87)90125-7. URL https://www.sciencedirect.com/science/article/pii/0377042787901257.

[45] Engelmore R.S. Artificial intelligence and knowledge based systems: Origins, methods and opportunities for nde. in: Thompson d.o., chimenti d.e. (eds) review of progress in quantitative nondestructive evaluation. review of progress in quantitative nondestructive evaluation. *A. Springer, Boston, MA*, vol 6, 1987. doi: https://doi.org/10.1007/978-1-4613-1893-4_1.

[46] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall Press, USA, 3rd edition, 2009. ISBN 0136042597.

[47] de Vries P.H Stokman F.N. Structuring knowledge in a graph. in: van der veer g.c., mulder g. (eds) human-computer interaction. springer, berlin, heidelberg. 1988. doi: https://doi.org/10.1007/978-3-642-73402-1_12.

[48] Yonatan Tesfaye, Eyasu Mequanint, Marcello Pelillo, and Andrea Prati. Multi-object tracking using dominant sets. *IET Computer Vision*, 10, 03 2016. doi: 10.1049/iet-cvi.2015.0297.

[49] K. Tripathi. A review on knowledge-based expert system: Concept and architecture. ijca special issue on artificial intelligence techniques-novel approaches & practical applications. 2011. doi: https://doi.org/10.5120/2845-226.

[50] Sebastiano Vascon, E. Mequanint, Marco Cristani, Hayley Hung, M. Pelillo, and Vittorio Murino. Detecting conversational groups in images and sequences: A robust game-theoretic approach. *Comput. Vis. Image Underst.*, 143:11–24, 2016.

[51] Ulrike von Luxburg. A tutorial on spectral clustering, 2007.

[52] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, AAAI'14, page 1112–1119. AAAI Press, 2014.

[53] Jorgen W. Weibull. *Evolutionary Game Theory*, volume 1 of *MIT Press Books*. The MIT Press, September 1997. ISBN ARRAY(0x44753738). URL https://ideas.repec.org/b/mtp/titles/0262731215.html.

[54] Jörgen W. Weibull. *Evolutionary game theory*. MIT Press, Cambridge, Mass. [u.a.], 1995. ISBN 0262231816. URL http://gso.gbv.de/DB=2.1/CMD?ACT=SRCHA&SRT=YOP&IKT=1016&TRM=ppn+180190040&sourceid=fbw_bibsonomy.

[55] E.A. Bender S.G. Williamson. Lists, decisions and graphs. with an introduction to probability. *University of California at San Diego*, page 108, 2010. doi: https://cseweb.ucsd.edu/~gill/BWLectSite/Resources/LDGbookCOV.pdf.

[56] Eyasu Zemene, Yonatan Tariku, Haroon Idrees, Andrea Prati, Marcello Pelillo, and Mubarak Shah. Large-scale image geo-localization using dominant sets, 2017.

Appendix A

Tables Results

A.1 Graph Clustering

Dominant set clustering							
Settings		Cut off 1.0e-4			Cut off 1.0e-8		
Dataset	Hops	Clusters	Modularity	Avg. Coherence	Clusters	Modularity	Avg. Coherence
Barcelona	2	225	**0.159**	0.592	202	0.146	0.613
Barcelona	3	394	0.081	0.692	355	**0.137**	0.695
Lisbon	2	99	0.187	0.576	90	**0.207**	0.594
Lisbon	3	305	0.070	0.836	248	**0.133**	0.847

Table A.1: Dominant Set clustering with weighted adj. matrix pilot cities

Dominant set clustering							
Settings		Cut off 1.0e-4			Cut off 1.0e-8		
Dataset	Hops	Clusters	Modularity	Avg. Coherence	Clusters	Modularity	Avg. Coherence
Barcelona_R	2	137	0.322	0.582	127	**0.368**	0.569
Barcelona_R	3	483	0.161	0.795	324	**0.218**	0.819
Lisbon_R	2	56	0.426	0.567	54	**0.494**	0.548
Lisbon_R	3	250	0.058	0.592	250	**0.205**	0.606

Table A.2: Dominant Set clustering with weighted adj. matrix reduced pilot cities

Spectral Clustering							
Barcelona				Lisbon			
2 Hops		3 Hops		2 Hops		3 Hops	
Clusters	Modularity	Clusters	Modularity	Clusters	Modularity	Clusters	Modularity
9	0.134	36	0.209	12	0.177	6	0.326
36	0.281	11	0.118	11	**0.304**	4	0.008
34	**0.299**	22	0.166	22	0.256	12	0.384
27	0.217	27	**0.316**	24	0.228	3	0.072
25	0.209	66	0.292	35	0.272	14	**0.468**

Table A.3: Spectral clustering with weighted adj. matrix for Barcelona and Lisbon

Spectral Clustering							
Barcelona_R				Lisbon_R			
2 Hops		3 Hops		2 Hops		3 Hops	
Clusters	Modularity	Clusters	Modularity	Clusters	Modularity	Clusters	Modularity
194	0.399	98	0.366	147	0.511	50	0.372
203	**0.605**	106	0.484	167	**0.564**	73	0.483
190	0.239	96	**0.502**	148	0.434	86	0.473
216	0.283	102	0.326	175	0.421	63	**0.496**
212	0.217	122	0.290	213	0.403	60	0.412

Table A.4: Spectral clustering with weighted adj. matrix for Barcelona and Lisbon reduced

Louvain Community			
Settings		Results	
Dataset	Hops	Clusters	Modularity
Barcelona	2	46	0.462
Barcelona	3	32	0.564
Barcelona_R	2	243	0.742
Barcelona_R	3	147	0.778
Lisbon	2	46	0.479
Lisbon	3	33	0.678
Lisbon_R	2	158	0.828
Lisbon_R	3	88	0.819
Paris	2	34	0.526
Paris_R	2	802	0.766

Table A.5: Louvain community with pilot cities

A.2 Topology Embedding

A.2.1 DeepWalk

Dominant Set - DeepWalk											
Lisbon 2H						Lisbon 2HR					
cutoff	Clusters	Modularity	Silhouette	Davies	CHI	cutoff	Clusters	Modularity	Silhouette	Davies	CHI
0.0001	511	0.254	0.224	1.448	16.021	0.0001	282	0.613	0.370	1.336	30.145
1.00E-08	309	0.262	0.131	1.999	14.080	1.00E-08	175	0.610	0.241	1.804	24.301
1.00E-16	133	0.277	-0.009	2.681	11.590	1.00E-16	82	0.619	0.164	2.267	27.885

Table A.6: Dominant Set - DeepWalk

DBSCAN - DeepWalk											
Lisbon 2H						Lisbon 2HR					
Epsilon	Clusters	Modularity	Silhouette	Davies	CHI	Epsilon	Clusters	Modularity	Silhouette	Davies	CHI
2	201	0.174	0.052	1.337	9.969	2	214	0.518	0.269	1.173	13.106
2.5	237	0.278	0.165	1.397	11.714	3	194	0.645	0.253	1.197	14.491
3	101	0.155	0.087	1.464	12.458	3	119	0.495	0.146	1.028	13.063

Table A.7: DBSCAN - DeepWalk

Spectral Clustering - DeepWalk									
Lisbon 2H					Lisbon 2HR				
Clusters	Modularity	Silhouette	Davies	CHI	Clusters	Modularity	Silhouette	Davies	CHI
131	0.362	0.149	1.980	20.919	129	0.717	0.281	1.638	24.947
128	0.378	0.157	2.014	21.444	92	0.754	0.267	1.681	31.889
122	0.391	0.148	2.064	21.022	60	0.777	0.234	1.666	40.257

Table A.8: Spectral Clustering - DeepWalk

K-Means - DeepWalk									
Lisbon 2H					Lisbon 2HR				
Clusters	Modularity	Silhouette	Davies	CHI	Clusters	Modularity	Silhouette	Davies	CHI
131	0.395	0.176	1.777	22.908	129	0.763	0.310	1.261	32.469
128	0.402	0.166	1.777	22.882	92	0.781	0.280	1.355	36.683
122	0.394	0.178	1.825	23.368	60	0.787	0.244	1.613	41.662

Table A.9: K-Means - DeepWalk

A.2.2 Node2Vec

Dominant Set -Node2vec											
Lisbon 2H						Lisbon 2HR					
cutoff	Clusters	Modularity	Silhouette	Davies	CHI	cutoff	Clusters	Modularity	Silhouette	Davies	CHI
0.0001	479	0.255	0.224	1.587	14.218	0.0001	320	0.579	0.458	1.026	42.475
1.00E-08	285	0.264	0.139	2.046	13.666	1.00E-08	140	0.630	0.227	2.057	22.855
1.00E-16	121	0.309	0.038	2.691	15.313	1.00E-16	60	0.660	0.167	2.465	31.726

Table A.10: Dominant Set -Node2vec

DBSCAN - Node2vec											
Lisbon 2H						Lisbon 2HR					
Epsilon	Clusters	Modularity	Silhouette	Davies	CHI	Epsilon	Clusters	Modularity	Silhouette	Davies	CHI
1.5	78	0.044	-0.054	1.187	9.292	1.5	216	0.534	0.291	1.252	14.736
2	316	0.221	0.102	1.444	8.649	2	167	0.683	0.240	1.379	14.532
2.5	126	0.182	-0.001	1.738	7.876	2.5	66	0.215	0.003	1.233	6.884
3	23	0.006	0.019	1.476	8.294	3	54	0.185	0.149	1.196	7.411

Table A.11: DBSCAN - Node2vec

Spectral Clustering - Node2Vec									
Lisbon 2H					Lisbon 2HR				
Clusters	Modularity	Silhouette	Davies	CHI	Clusters	Modularity	Silhouette	Davies	CHI
96	0.402	0.144	2.060	23.906	91	0.775	0.280	1.493	28.102
85	0.406	0.132	2.196	24.918	65	0.783	0.261	1.719	35.321
78	0.411	0.133	2.277	26.237	59	0.786	0.255	1.807	38.551

Table A.12: Spectral Clustering - Node2Vec

K-Means - Node2Vec									
Lisbon 2H					Lisbon 2HR				
Clusters	Modularity	Silhouette	Davies	CHI	Clusters	Modularity	Silhouette	Davies	CHI
96	0.412	0.151	2.075	25.231	91	0.763	0.235	1.603	29.741
85	0.422	0.144	2.132	26.129	65	0.773	0.208	1.751	33.196
78	0.426	0.143	2.099	27.317	59	0.785	0.173	1.717	34.209

Table A.13: K-Means - Node2Vec

A.3 Semantics Embeddings

A.3.1 FastText

Dominant Set -FastText											
Lisbon 2H						**Lisbon 2HR**					
cutoff	**Clusters**	**Modularity**	**Silhouette**	**Davies**	**CHI**	**cutoff**	**Clusters**	**Modularity**	**Silhouette**	**Davies**	**CHI**
0.0001	170	-0.049	0.591	2.128	141.434	0.0001	128	0.002	0.631	1.952	149.939
1.00E-08	89	-0.048	0.503	2.510	170.720	1.00E-08	67	0.000	0.552	2.376	170.164
1.00E-12	12	-0.034	0.024	2.954	38.472	1.00E-12	6	-0.014	0.073	2.601	39.136

Table A.14: Dominant Set -FastText

DBSCAN - FastText											
Lisbon 2H						**Lisbon 2HR**					
Epsilon	**Clusters**	**Modularity**	**Silhouette**	**Davies**	**CHI**	**Epsilon**	**Clusters**	**Modularity**	**Silhouette**	**Davies**	**CHI**
0.3	73	-0.091	0.540	0.931	162.019	0.3	58	0.028	0.582	0.965	160.381
0.5	71	-0.062	0.546	1.006	176.829	0.5	62	0.026	0.602	1.076	161.543
1	14	-0.062	0.432	1.635	446.672	1	10	-0.009	0.466	1.677	491.663

Table A.15: DBSCAN - FastText

Spectral Clustering - FastText									
Lisbon 2H					**Lisbon 2HR**				
Clusters	**Modularity**	**Silhouette**	**Davies**	**CHI**	**Clusters**	**Modularity**	**Silhouette**	**Davies**	**CHI**
96	-0.026	0.531	2.326	167.700	94	0.030	0.587	1.941	156.779
80	-0.024	0.526	2.235	196.529	84	0.019	0.570	2.121	153.278
69	-0.028	0.524	2.392	217.784	80	0.029	0.583	2.078	168.549

Table A.16: Spectral Clustering - FastText

K-Means - FastText									
Lisbon 2H					**Lisbon 2HR**				
Clusters	**Modularity**	**Silhouette**	**Davies**	**CHI**	**Clusters**	**Modularity**	**Silhouette**	**Davies**	**CHI**
96	-0.047	0.580	1.759	228.115	94	0.004	0.631	1.446	212.980
80	-0.042	0.571	1.719	257.167	84	0.010	0.628	1.503	220.875
69	-0.033	0.572	1.723	283.031	80	0.013	0.628	1.441	233.025

Table A.17: K-Means - FastText

A.3.2 Word2Vec

Dominant Set -Word2Vec											
Lisbon 2H						**Lisbon 2HR**					
cutoff	**Clusters**	**Modularity**	**Silhouette**	**Davies**	**CHI**	**cutoff**	**Clusters**	**Modularity**	**Silhouette**	**Davies**	**CHI**
0.0001	186	-0.048	0.609	1.937	133.516	0.0001	137	-0.001	0.650	1.872	149.248
1.00E-08	93	-0.048	0.485	2.379	124.717	1.00E-08	62	0.000	0.526	2.238	138.260
1.00E-12	5	-0.018	0.204	2.376	66.691	1.00E-12	4	-0.017	0.285	1.829	97.108

Table A.18: Dominant Set -Word2Vec

DBSCAN - Word2Vec											
Lisbon 2H						**Lisbon 2HR**					
Epsilon	**Clusters**	**Modularity**	**Silhouette**	**Davies**	**CHI**	**Epsilon**	**Clusters**	**Modularity**	**Silhouette**	**Davies**	**CHI**
0.3	75	-0.092	0.542	1.027	155.478	0.3	60	0.030	0.588	1.050	156.006
0.5	76	-0.058	0.544	1.036	156.255	0.5	62	0.026	0.590	1.064	154.229
1	51	-0.068	0.422	1.204	188.684	1	43	0.018	0.482	1.261	183.120

Table A.19: DBSCAN - Word2Vec

Spectral Clustering - Word2Vec									
Lisbon 2H					Lisbon 2HR				
Clusters	Modularity	Silhouette	Davies	CHI	Clusters	Modularity	Silhouette	Davies	CHI
102	-0.025	0.538	2.140	159.280	99	0.028	0.600	1.362	202.819
99	-0.031	0.539	2.131	164.173	91	0.031	0.586	1.391	203.067
64	-0.017	0.518	2.243	223.923	74	0.033	0.580	1.355	223.712

Table A.20: Spectral Clustering - Word2Vec

K-Means - Word2Vec									
Lisbon 2H					Lisbon 2HR				
Clusters	Modularity	Silhouette	Davies	CHI	Clusters	Modularity	Silhouette	Davies	CHI
102	-0.040	0.579	1.536	206.569	99	0.019	0.632	1.893	154.788
99	-0.040	0.575	1.578	210.758	91	0.014	0.617	1.911	157.815
64	-0.039	0.556	1.626	271.303	74	0.029	0.620	2.036	174.170

Table A.21: K-Means - Word2Vec

A.4 Translational Embeddings

Dominant Set -TransE											
Lisbon 2H						Lisbon 2HR					
cutoff	Clusters	Modularity	Silhouette	Davies	CHI	cutoff	Clusters	Modularity	Silhouette	Davies	CHI
0.0001	556	0.014	0.107	1.827	5.914	0.0001	451	0.070	0.099	1.850	4.012
1.00E-08	319	0.011	0.057	2.450	6.370	1.00E-08	247	0.053	0.039	2.640	4.111
1.00E-12	174	0.008	0.028	2.917	7.765	1.00E-12	146	0.048	0.010	3.145	4.474

Table A.22: Dominant Set -TransE

DBSCAN - TransE											
Lisbon 2H						Lisbon 2HR					
Epsilon	Clusters	Modularity	Silhouette	Davies	CHI	Epsilon	Clusters	Modularity	Silhouette	Davies	CHI
0.3	1	-0.018	None	None	None	0.3	1	-0.014	None	None	None
0.5	7	-0.018	-0.056	1.414	3.416	0.5	1	-0.014	None	None	None
1	45	0.005	-0.070	2.254	3.431	1	118	0.080	-0.025	2.360	3.134

Table A.23: DBSCAN - TransE

Spectral Clustering - TransE									
Lisbon 2H					Lisbon 2HR				
Clusters	Modularity	Silhouette	Davies	CHI	Clusters	Modularity	Silhouette	Davies	CHI
107	0.026	0.068	2.789	11.351	103	0.071	0.041	3.087	6.271
105	0.025	0.065	2.784	11.158	101	0.073	0.032	3.132	6.195
103	0.028	0.068	2.742	11.372	100	0.070	0.034	3.203	6.254

Table A.24: Spectral Clustering - TransE

K-Means - TransE									
Lisbon 2H					Lisbon 2HR				
Clusters	Modularity	Silhouette	Davies	CHI	Clusters	Modularity	Silhouette	Davies	CHI
107	0.017	0.057	3.025	11.697	103	0.064	0.037	3.288	6.089
105	0.021	0.061	2.983	11.972	101	0.062	0.038	3.278	6.174
103	0.019	0.059	3.092	11.794	100	0.069	0.037	3.288	6.188

Table A.25: K-Means - TransE

A.5 Embedding Combination

A.5.1 Topology Embeddings and Description Embeddings

DeepWalk and FastText

Dominant Set -DeepWalk + FastText											
Lisbon 2H						Lisbon 2HR					
cutoff	Clusters	Modularity	Silhouette	Davies	CHI	cutoff	Clusters	Modularity	Silhouette	Davies	CHI
0.0001	256	0.002	-0.060	2.977	7.512	0.0001	199	0.074	0.000	2.720	9.323
1.00E-08	154	0.006	-0.049	3.525	9.100	1.00E-08	124	0.052	-0.029	3.144	9.087
1.00E-16	66	-0.010	-0.031	4.410	13.650	1.00E-16	51	0.074	-0.030	3.839	12.317

Table A.26: Dominant Set -DeepWalk + FastText

DBSCAN - DeepWalk + FastText											
Lisbon 2H						Lisbon 2HR					
Epsilon	Clusters	Modularity	Silhouette	Davies	CHI	Epsilon	Clusters	Modularity	Silhouette	Davies	CHI
1.5	44	-0.011	-0.244	1.618	4.259	1.5	78	0.064	-0.114	1.397	6.592
2	90	0.011	-0.161	1.587	6.843	2	120	0.245	-0.001	1.331	7.932
2.5	143	0.095	-0.004	1.498	9.227	2.5	154	0.402	0.109	1.377	9.761
3	142	0.175	0.051	1.554	10.104	3	145	0.543	0.134	1.300	11.405

Table A.27: DBSCAN - DeepWalk + FastText

Spectral Clustering - DeepWalk + FastText									
Lisbon 2H					Lisbon 2HR				
Clusters	Modularity	Silhouette	Davies	CHI	Clusters	Modularity	Silhouette	Davies	CHI
62	0.289	0.058	2.071	24.482	62	0.641	0.145	1.722	33.116
93	0.254	0.076	2.133	21.607	56	0.647	0.136	1.759	34.722
128	0.272	0.101	2.110	19.637	53	0.637	0.124	1.803	33.715

Table A.28: Spectral Clustering - DeepWalk + FastText

K-Means - DeepWalk + FastText									
Lisbon 2H					Lisbon 2HR				
Clusters	Modularity	Silhouette	Davies	CHI	Clusters	Modularity	Silhouette	Davies	CHI
62	0.251	0.104	2.207	27.775	62	0.689	0.179	1.781	35.237
93	0.237	0.125	2.090	24.051	56	0.690	0.167	1.801	37.690
128	0.255	0.122	1.938	21.435	53	0.680	0.138	1.859	35.444

Table A.29: K-Means - DeepWalk + FastText

DeepWalk and Word2Vec

Dominant Set -DeepWalk + Word2Vec											
Lisbon 2H						Lisbon 2HR					
cutoff	Clusters	Modularity	Silhouette	Davies	CHI	cutoff	Clusters	Modularity	Silhouette	Davies	CHI
0.0001	262	0.010	-0.020	2.726	9.725	0.0001	207	0.067	0.040	2.536	10.565
1.00E-08	170	0.003	-0.026	3.170	11.686	1.00E-08	123	0.085	0.010	2.971	11.207
1.00E-16	66	-0.015	-0.031	4.382	18.210	1.00E-16	57	0.075	0.002	3.446	15.925

Table A.30: Dominant Set -DeepWalk + Word2Vec

DBSCAN - DeepWalk + Word2Vec											
Lisbon 2H						Lisbon 2HR					
Epsilon	Clusters	Modularity	Silhouette	Davies	CHI	Epsilon	Clusters	Modularity	Silhouette	Davies	CHI
2	79	-0.010	-0.192	1.547	5.727	2	88	0.137	-0.073	1.398	7.029
2.5	98	-0.020	-0.078	1.506	9.141	2.5	110	0.292	0.016	1.422	8.571
3	114	0.121	0.019	1.558	11.881	3	102	0.415	0.058	1.459	10.631

Table A.31: DBSCAN - DeepWalk + Word2Vec

Spectral Clustering - DeepWalk + Word2Vec									
Lisbon 2H					Lisbon 2HR				
Clusters	Modularity	Silhouette	Davies	CHI	Clusters	Modularity	Silhouette	Davies	CHI
61	0.213	0.032	2.035	26.517	42	0.626	0.120	1.836	34.518
59	0.196	0.031	2.141	27.503	39	0.595	0.116	1.891	35.344
54	0.170	0.029	1.970	29.016	87	0.547	0.162	1.816	28.280

Table A.32: Spectral Clustering - DeepWalk + Word2Vec

K-Means - DeepWalk + Word2Vec									
Lisbon 2H					Lisbon 2HR				
Clusters	Modularity	Silhouette	Davies	CHI	Clusters	Modularity	Silhouette	Davies	CHI
61	0.175	0.057	2.427	32.013	42	0.548	0.151	2.067	40.153
59	0.167	0.062	2.420	32.076	39	0.544	0.137	2.106	39.997
54	0.175	0.040	2.521	32.763	87	0.542	0.193	1.718	29.536

Table A.33: K-Means - DeepWalk + Word2Vec

Node2Vec and FastText

Dominant Set -Node2Vec + FastText											
Lisbon 2H						Lisbon 2HR					
cutoff	Clusters	Modularity	Silhouette	Davies	CHI	cutoff	Clusters	Modularity	Silhouette	Davies	CHI
0.0001	244	0.007	-0.014	2.932	8.732	0.0001	186	0.074	0.042	2.803	10.615
1.00E-08	158	0.001	-0.028	3.345	10.173	1.00E-08	114	0.057	-0.017	3.157	11.418
1.00E-16	60	-0.006	-0.017	4.237	17.776	1.00E-16	60	0.081	-0.025	3.511	14.551

Table A.34: Dominant Set -Node2Vec + FastText

DBSCAN - Node2Vec + FastText											
Lisbon 2H						Lisbon 2HR					
Epsilon	Clusters	Modularity	Silhouette	Davies	CHI	Epsilon	Clusters	Modularity	Silhouette	Davies	CHI
1.5	22	-0.014	-0.045	1.177	8.792	1.5	107	0.147	-0.046	1.313	8.265
2	110	0.054	-0.133	1.431	6.732	2	149	0.362	0.100	1.418	10.685
2.5	205	0.100	0.033	1.631	8.951	2.5	103	0.340	0.029	1.530	9.798
3	51	-0.008	-0.058	1.972	6.237	3	56	0.186	0.051	1.492	6.270

Table A.35: DBSCAN - Node2Vec + FastText

Spectral Clustering - Node2Vec + FastText									
Lisbon 2H					Lisbon 2HR				
Clusters	Modularity	Silhouette	Davies	CHI	Clusters	Modularity	Silhouette	Davies	CHI
55	0.256	0.055	2.272	26.867	83	0.540	0.138	2.155	24.518
149	0.234	0.106	2.242	18.439	35	0.598	0.111	2.240	37.656
62	0.273	0.054	2.205	24.933	37	0.578	0.111	2.244	36.669

Table A.36: Spectral Clustering - Node2Vec + FastText

K-Means - Node2Vec + FastText									
Lisbon 2H					Lisbon 2HR				
Clusters	Modularity	Silhouette	Davies	CHI	Clusters	Modularity	Silhouette	Davies	CHI
55	0.281	0.079	2.580	31.452	83	0.597	0.194	1.926	29.163
149	0.243	0.120	2.204	19.858	35	0.586	0.114	2.153	40.958
62	0.282	0.095	2.746	29.832	37	0.637	0.154	2.282	42.935

Table A.37: K-Means - Node2Vec + FastText

Node2Vec and Word2Vec

Dominant Set -Node2Vec + Word2Vec											
Lisbon 2H						Lisbon 2HR					
cutoff	Clusters	Modularity	Silhouette	Davies	CHI	cutoff	Clusters	Modularity	Silhouette	Davies	CHI
0.0001	264	0.014	0.019	2.672	11.586	0.0001	196	0.076	0.089	2.531	13.501
1.00E-08	170	0.009	0.004	2.974	14.150	1.00E-08	126	0.083	0.024	2.829	14.287
1.00E-16	76	-0.003	0.002	3.646	23.095	1.00E-16	53	0.052	0.037	3.431	22.780

Table A.38: Dominant Set -Node2Vec + Word2Vec

DBSCAN - Node2Vec + Word2Vec											
Lisbon 2H						Lisbon 2HR					
Epsilon	Clusters	Modularity	Silhouette	Davies	CHI	Epsilon	Clusters	Modularity	Silhouette	Davies	CHI
2	85	0.015	-0.192	1.383	5.831	2	104	0.225	-0.001	1.384	9.958
2.5	133	0.040	-0.067	1.624	8.940	2.5	89	0.330	0.035	1.508	12.259
3	93	-0.054	-0.008	1.760	13.345	3	58	0.216	0.040	1.532	14.493

Table A.39: DBSCAN - Node2Vec + Word2Vec

Spectral Clustering - Node2Vec + Word2Vec									
Lisbon 2H					Lisbon 2HR				
Clusters	Modularity	Silhouette	Davies	CHI	Clusters	Modularity	Silhouette	Davies	CHI
61	0.189	0.058	2.499	33.629	98	0.440	0.129	2.060	24.429
80	0.210	0.062	2.572	29.324	62	0.471	0.123	2.244	30.923
72	0.192	0.059	2.476	31.628	49	0.488	0.129	2.188	35.044

Table A.40: Spectral Clustering - Node2Vec + Word2Vec

K-Means - Node2Vec + Word2Vec									
Lisbon 2H					Lisbon 2HR				
Clusters	Modularity	Silhouette	Davies	CHI	Clusters	Modularity	Silhouette	Davies	CHI
61	0.181	0.073	2.761	37.831	98	0.449	0.200	2.100	28.889
80	0.190	0.078	2.510	31.435	62	0.435	0.158	2.203	36.321
72	0.189	0.071	2.755	33.236	49	0.438	0.149	2.496	41.491

Table A.41: K-Means - Node2Vec + Word2Vec

A.5.2 Topology Embeddings and Translational Embeddings

DeepWalk and TransE

Dominant Set -DeepWalk + TransE											
Lisbon 2H						Lisbon 2HR					
cutoff	**Clusters**	**Modularity**	**Silhouette**	**Davies**	**CHI**	**cutoff**	**Clusters**	**Modularity**	**Silhouette**	**Davies**	**CHI**
0.0001	605	0.139	0.082	1.933	8.124	0.0001	372	0.392	0.171	1.745	17.500
1.00E-08	368	0.128	0.040	2.461	8.065	1.00E-08	231	0.414	0.123	2.139	17.469
1.00E-16	113	0.128	-0.060	3.687	8.083	1.00E-16	106	0.533	0.094	2.480	18.775

Table A.42: Dominant Set -DeepWalk + TransE

DBSCAN - DeepWalk + TransE											
Lisbon 2H						Lisbon 2HR					
Epsilon	**Clusters**	**Modularity**	**Silhouette**	**Davies**	**CHI**	**Epsilon**	**Clusters**	**Modularity**	**Silhouette**	**Davies**	**CHI**
2	78	0.019	-0.194	1.754	6.191	2	154	0.417	0.091	1.245	10.218
2.5	230	0.198	0.075	1.421	9.492	2.5	217	0.532	0.211	1.269	12.346
3	204	0.260	0.137	1.487	11.015	3	170	0.615	0.190	1.243	13.540

Table A.43: DBSCAN - DeepWalk + TransE

Spectral Clustering - DeepWalk + TransE									
Lisbon 2H					Lisbon 2HR				
Clusters	**Modularity**	**Silhouette**	**Davies**	**CHI**	**Clusters**	**Modularity**	**Silhouette**	**Davies**	**CHI**
98	0.366	0.097	2.103	20.509	46	0.769	0.176	1.727	37.910
78	0.372	0.091	2.178	21.918	78	0.749	0.182	1.618	32.686
85	0.365	0.092	2.211	21.482	47	0.770	0.178	1.718	37.695

Table A.44: Spectral Clustering - DeepWalk + TransE

K-Means - DeepWalk + TransE									
Lisbon 2H					Lisbon 2HR				
Clusters	**Modularity**	**Silhouette**	**Davies**	**CHI**	**Clusters**	**Modularity**	**Silhouette**	**Davies**	**CHI**
98	0.432	0.147	1.942	22.439	46	0.794	0.178	1.863	38.776
78	0.415	0.112	2.097	22.868	78	0.775	0.217	1.672	33.325
85	0.405	0.122	2.062	22.068	47	0.795	0.178	1.872	40.400

Table A.45: K-Means - DeepWalk + TransE

Node2Vec and TransE

Dominant Set -Node2Vec + TransE											
Lisbon 2H						Lisbon 2HR					
cutoff	**Clusters**	**Modularity**	**Silhouette**	**Davies**	**CHI**	**cutoff**	**Clusters**	**Modularity**	**Silhouette**	**Davies**	**CHI**
0.0001	587	0.152	0.095	1.921	7.641	0.0001	378	0.436	0.229	1.661	20.060
1.00E-08	355	0.158	0.053	2.406	8.117	1.00E-08	177	0.449	0.079	2.531	14.768
1.00E-16	106	0.172	-0.020	3.296	10.662	1.00E-16	77	0.513	0.059	2.736	19.050

Table A.46: Dominant Set -Node2Vec + TransE

DBSCAN - Node2Vec + TransE											
Lisbon 2H						Lisbon 2HR					
Epsilon	**Clusters**	**Modularity**	**Silhouette**	**Davies**	**CHI**	**Epsilon**	**Clusters**	**Modularity**	**Silhouette**	**Davies**	**CHI**
2	118	0.071	-0.059	1.330	6.995	2	212	0.518	0.175	1.374	11.867
2.5	316	0.235	0.109	1.618	8.469	2.5	148	0.677	0.114	1.551	10.163
3	54	0.054	-0.042	1.904	6.735	3	60	0.199	0.018	1.384	6.594

Table A.47: DBSCAN - Node2Vec + TransE

Spectral Clustering - Node2Vec + TransE									
Lisbon 2H					Lisbon 2HR				
Clusters	Modularity	Silhouette	Davies	CHI	Clusters	Modularity	Silhouette	Davies	CHI
69	0.423	0.090	2.350	21.226	75	0.737	0.134	1.997	24.003
57	0.408	0.073	2.219	21.631	68	0.749	0.132	2.024	25.511
64	0.390	0.080	2.206	20.850	60	0.755	0.129	2.080	27.120

Table A.48: Spectral Clustering - Node2Vec + TransE

K-Means - Node2Vec + TransE									
Lisbon 2H					Lisbon 2HR				
Clusters	Modularity	Silhouette	Davies	CHI	Clusters	Modularity	Silhouette	Davies	CHI
69	0.431	0.113	2.393	24.821	75	0.775	0.158	1.820	26.388
57	0.432	0.100	2.486	25.649	68	0.787	0.183	1.922	29.120
64	0.434	0.100	2.499	24.563	60	0.791	0.187	1.960	31.506

Table A.49: K-Means - Node2Vec + TransE

A.5.3 Description Embeddings and Translational Embeddings

FastText and TransE

Dominant Set -FastText + TransE											
Lisbon 2H						Lisbon 2HR					
cutoff	Clusters	Modularity	Silhouette	Davies	CHI	cutoff	Clusters	Modularity	Silhouette	Davies	CHI
0.0001	195	-0.038	0.026	2.788	24.189	0.0001	144	0.012	0.166	2.734	23.390
1.00E-08	128	-0.020	0.026	3.354	34.162	1.00E-08	87	0.009	0.175	3.198	34.925
1.00E-12	87	-0.024	0.036	3.571	46.721	1.00E-12	24	0.006	-0.002	3.403	15.821

Table A.50: Dominant Set -FastText + TransE

DBSCAN - FastText + TransE											
Lisbon 2H						Lisbon 2HR					
Epsilon	Clusters	Modularity	Silhouette	Davies	CHI	Epsilon	Clusters	Modularity	Silhouette	Davies	CHI
1	66	-0.023	-0.228	1.877	11.695	1	48	-0.012	-0.285	1.553	4.574
1.5	21	-0.046	0.129	2.149	106.503	1.5	25	0.033	0.146	2.029	81.562
2	4	-0.081	0.383	1.990	616.994	2	4	-0.013	0.377	2.337	456.184

Table A.51: DBSCAN - FastText + TransE

Spectral Clustering - FastText + TransE									
Lisbon 2H					Lisbon 2HR				
Clusters	Modularity	Silhouette	Davies	CHI	Clusters	Modularity	Silhouette	Davies	CHI
186	0.008	-0.011	2.808	21.118	163	0.044	-0.055	1.315	13.202
163	0.006	-0.002	2.857	23.798	159	0.021	-0.051	1.345	13.207
149	-0.001	0.004	2.909	25.774	154	0.011	-0.050	1.365	13.494

Table A.52: Spectral Clustering - FastText + TransE

K-Means - FastText + TransE									
Lisbon 2H					Lisbon 2HR				
Clusters	Modularity	Silhouette	Davies	CHI	Clusters	Modularity	Silhouette	Davies	CHI
186	-0.002	0.032	3.135	26.546	163	0.030	0.027	3.027	21.588
163	-0.008	0.039	3.182	29.735	159	0.008	0.027	3.021	22.062
149	-0.004	0.035	3.330	31.583	154	0.011	0.029	3.078	22.716

Table A.53: K-Means - FastText + TransE

Word2Vec and TransE

Dominant Set -Word2Vec + TransE											
Lisbon 2H						Lisbon 2HR					
cutoff	Clusters	Modularity	Silhouette	Davies	CHI	cutoff	Clusters	Modularity	Silhouette	Davies	CHI
0.0001	205	-0.038	0.069	2.494	39.699	0.0001	149	0.014	0.261	2.371	41.847
1.00E-08	147	-0.019	0.070	2.936	50.541	1.00E-08	98	0.004	0.248	2.724	55.276
1.00E-12	90	-0.025	0.053	3.302	70.204	1.00E-12	52	-0.019	0.132	2.902	31.205

Table A.54: Dominant Set -Word2Vec + TransE

DBSCAN - Word2Vec + TransE											
Lisbon 2H						Lisbon 2HR					
Epsilon	Clusters	Modularity	Silhouette	Davies	CHI	Epsilon	Clusters	Modularity	Silhouette	Davies	CHI
1	63	-0.022	-0.295	1.841	13.626	1	45	-0.011	-0.372	1.506	5.097
1.5	43	-0.059	0.224	1.596	112.362	1.5	46	0.036	0.229	1.766	85.982
2	15	-0.058	0.294	2.176	255.698	2	13	0.001	0.258	2.237	228.877

Table A.55: DBSCAN - Word2Vec + TransE

Spectral Clustering - Word2Vec + TransE									
Lisbon 2H					Lisbon 2HR				
Clusters	Modularity	Silhouette	Davies	CHI	Clusters	Modularity	Silhouette	Davies	CHI
159	0.000	0.000	2.744	38.791	188	0.032	0.014	1.574	25.940
151	0.000	0.005	2.801	40.440	157	0.035	0.013	1.2	31.385
78	-0.005	0.078	2.937	76.237	152	0.028	-0.02	1.35	32.068

Table A.56: Spectral Clustering - Word2Vec + TransE

K-Means - Word2Vec + TransE									
Lisbon 2H					Lisbon 2HR				
Clusters	Modularity	Silhouette	Davies	CHI	Clusters	Modularity	Silhouette	Davies	CHI
159	-0.003	0.049	2.744	51.041	188	0.018	0.046	2.443	36.126
151	-0.006	0.049	2.777	53.313	157	0.021	0.051	2.560	41.253
78	0.001	0.054	2.915	90.111	152	0.015	0.044	2.578	42.050

Table A.57: K-Means - Word2Vec + TransE

A.5.4 Topology - Description - Translational

DeepWalk - FastText - TransE

Dominant Set - DeepWalk + FastText + TransE											
Lisbon 2H						Lisbon 2HR					
cutoff	Clusters	Modularity	Silhouette	Davies	CHI	cutoff	Clusters	Modularity	Silhouette	Davies	CHI
0.0001	264	0.006	-0.056	3.050	6.865	0.0001	202	0.078	0.001	2.775	8.767
1.00E-08	168	0.006	-0.056	3.532	8.219	1.00E-08	132	0.066	-0.036	3.189	8.541
1.00E-16	67	-0.011	-0.061	4.871	11.354	1.00E-16	59	0.080	-0.038	3.959	11.951

Table A.58: Dominant Set - DeepWalk + FastText + TransE

DBSCAN - DeepWalk + FastText + TransE											
Lisbon 2H						Lisbon 2HR					
Epsilon	Clusters	Modularity	Silhouette	Davies	CHI	Epsilon	Clusters	Modularity	Silhouette	Davies	CHI
2	52	-0.011	-0.224	1.713	4.632	2	72	0.085	-0.125	1.492	6.790
2.5	105	0.022	-0.120	1.655	7.180	2.5	130	0.289	-0.006	1.432	7.845
3	171	0.140	0.031	1.540	9.302	3	149	0.495	0.098	1.422	9.982

Table A.59: DBSCAN - DeepWalk + FastText + TransE

Spectral Clustering - DeepWalk + FastText + TransE									
Lisbon 2H					Lisbon 2HR				
Clusters	Modularity	Silhouette	Davies	CHI	Clusters	Modularity	Silhouette	Davies	CHI
193	0.244	0.122	2.057	15.628	137	0.563	0.201	1.763	24.169
52	0.213	0.061	2.011	25.034	52	0.630	0.113	1.933	31.902
63	0.255	0.049	2.162	22.855	70	0.616	0.137	1.900	29.334

Table A.60: Spectral Clustering - DeepWalk + FastText + TransE

K-Means - DeepWalk + FastText + TransE									
Lisbon 2H					Lisbon 2HR				
Clusters	Modularity	Silhouette	Davies	CHI	Clusters	Modularity	Silhouette	Davies	CHI
193	0.251	0.136	1.875	16.893	137	0.627	0.215	1.517	25.155
52	0.266	0.042	2.648	25.894	52	0.659	0.166	1.872	35.420
63	0.255	0.071	2.556	24.628	70	0.689	0.177	1.901	31.043

Table A.61: K-Means - DeepWalk + FastText + TransE

DeepWalk - Node2Vec - TransE

Dominant Set - DeepWalk + Word2Vec + TransE											
Lisbon 2H						Lisbon 2HR					
cutoff	Clusters	Modularity	Silhouette	Davies	CHI	cutoff	Clusters	Modularity	Silhouette	Davies	CHI
0.0001	282	0.012	-0.024	2.786	8.612	0.0001	211	0.073	0.028	2.553	9.922
1.00E-08	181	0.010	-0.038	3.226	10.281	1.00E-08	125	0.082	0.004	2.887	11.184
1.00E-16	92	0.005	-0.039	4.049	14.169	1.00E-16	60	0.088	-0.024	3.591	15.130

Table A.62: Dominant Set - DeepWalk + Word2Vec + TransE

DBSCAN - DeepWalk + Word2Vec + TransE											
Lisbon 2H						Lisbon 2HR					
Epsilon	Clusters	Modularity	Silhouette	Davies	CHI	Epsilon	Clusters	Modularity	Silhouette	Davies	CHI
2	49	-0.011	-0.245	1.623	3.752	2	64	0.031	-0.151	1.486	6.164
2.5	88	0.004	-0.166	1.615	6.425	2.5	91	0.198	-0.080	1.480	6.996
3	113	0.071	-0.052	1.592	9.182	3	104	0.328	0.005	1.485	8.956

Table A.63: DBSCAN - DeepWalk + Word2Vec + TransE

Spectral Clustering - DeepWalk + Word2Vec + TransE									
Lisbon 2H					Lisbon 2HR				
Clusters	Modularity	Silhouette	Davies	CHI	Clusters	Modularity	Silhouette	Davies	CHI
96	0.233	0.058	2.400	21.453	70	0.559	0.122	1.937	28.687
66	0.177	0.027	2.154	24.492	61	0.583	0.117	1.886	29.574
50	0.169	0.037	2.094	28.138	145	0.53	0.187	1.713	21.91

Table A.64: Spectral Clustering - DeepWalk + Word2Vec + TransE

K-Means - DeepWalk + Word2Vec + TransE									
Lisbon 2H					Lisbon 2HR				
Clusters	Modularity	Silhouette	Davies	CHI	Clusters	Modularity	Silhouette	Davies	CHI
96	0.194	0.077	2.350	23.339	70	0.520	0.129	2.009	28.623
66	0.173	0.044	2.626	26.672	61	0.527	0.143	1.998	32.398
50	0.185	0.048	2.652	33.057	145	0.506	0.202	1.643	22.290

Table A.65: K-Means - DeepWalk + Word2Vec + TransE

Node2Vec - FastText - TransE

Dominant Set - Node2Vec + FastText + TransE											
Lisbon 2H						Lisbon 2HR					
cutoff	Clusters	Modularity	Silhouette	Davies	CHI	cutoff	Clusters	Modularity	Silhouette	Davies	CHI
0.0001	254	0.009	-0.017	3.039	7.603	0.0001	192	0.068	0.005	2.881	9.028
1.00E-08	162	0.004	-0.032	3.539	8.998	1.00E-08	120	0.070	-0.005	3.175	10.392
1.00E-16	65	0.007	-0.033	4.395	15.764	1.00E-16	55	0.084	-0.001	3.853	16.071

Table A.66: Dominant Set - Node2Vec + FastText + TransE

DBSCAN - Node2Vec + FastText + TransE											
Lisbon 2H						Lisbon 2HR					
Epsilon	Clusters	Modularity	Silhouette	Davies	CHI	Epsilon	Clusters	Modularity	Silhouette	Davies	CHI
2	26	-0.014	-0.128	1.403	8.186	2	99	0.155	-0.062	1.419	8.039
2.5	158	0.085	-0.084	1.573	6.066	2.5	155	0.411	0.099	1.567	10.752
3	173	0.092	0.028	1.793	9.255	3	74	0.304	-0.023	1.598	6.678

Table A.67: DBSCAN - Node2Vec + FastText + TransE

Spectral Clustering - Node2Vec + FastText + TransE									
Lisbon 2H					Lisbon 2HR				
Clusters	Modularity	Silhouette	Davies	CHI	Clusters	Modularity	Silhouette	Davies	CHI
46	0.248	0.056	2.263	26.912	58	0.570	0.108	2.267	25.431
73	0.225	0.065	2.537	22.006	68	0.559	0.111	2.195	23.666
65	0.280	0.055	2.483	22.203	81	0.545	0.111	2.283	21.845

Table A.68: Spectral Clustering - Node2Vec + FastText + TransE

K-Means - Node2Vec + FastText + TransE									
Lisbon 2H					Lisbon 2HR				
Clusters	Modularity	Silhouette	Davies	CHI	Clusters	Modularity	Silhouette	Davies	CHI
46	0.252	0.072	3.182	30.373	58	0.584	0.135	2.186	28.982
73	0.248	0.085	2.655	24.195	68	0.575	0.152	2.152	28.300
65	0.231	0.072	2.748	25.363	81	0.598	0.158	2.072	25.438

Table A.69: K-Means - Node2Vec + FastText + TransE

Node2Vec - Word2Vec - TransE

Dominant Set -Node2Vec + Word2Vec + TransE											
Lisbon 2H						Lisbon 2HR					
cutoff	Clusters	Modularity	Silhouette	Davies	CHI	cutoff	Clusters	Modularity	Silhouette	Davies	CHI
0.0001	274	0.016	0.011	2.730	9.998	0.0001	203	0.080	0.048	2.550	11.616
1.00E-08	179	0.012	-0.008	3.212	12.218	1.00E-08	122	0.080	0.038	3.011	14.586
1.00E-16	81	-0.005	-0.025	3.769	19.727	1.00E-16	50	0.068	0.011	3.543	23.641

Table A.70: Dominant Set -Node2Vec + Word2Vec + TransE

DBSCAN - Node2Vec + Word2Vec + TransE											
Lisbon 2H						Lisbon 2HR					
Epsilon	Clusters	Modularity	Silhouette	Davies	CHI	Epsilon	Clusters	Modularity	Silhouette	Davies	CHI
2	21	-0.016	-0.161	1.254	6.512	2	78	0.056	-0.117	1.412	7.825
2.5	96	0.024	-0.160	1.568	5.931	2.5	100	0.258	-0.003	1.524	10.369
3	149	0.021	-0.023	1.764	9.301	3	83	0.267	0.010	1.620	10.739

Table A.71: DBSCAN - Node2Vec + Word2Vec + TransE

Spectral Clustering - Node2Vec + Word2Vec + TransE									
Lisbon 2H					Lisbon 2HR				
Clusters	Modularity	Silhouette	Davies	CHI	Clusters	Modularity	Silhouette	Davies	CHI
72	0.220	0.054	2.735	28.115	51	0.485	0.092	2.288	30.270
67	0.204	0.050	2.620	27.686	53	0.471	0.096	2.326	29.763
64	0.203	0.058	2.634	29.641	144	0.418	0.132	2.084	17.928

Table A.72: Spectral Clustering - Node2Vec + Word2Vec + TransE

K-Means - Node2Vec + Word2Vec + TransE									
Lisbon 2H					Lisbon 2HR				
Clusters	Modularity	Silhouette	Davies	CHI	Clusters	Modularity	Silhouette	Davies	CHI
72	0.192	0.066	2.921	29.283	51	0.434	0.121	2.455	35.427
67	0.189	0.069	2.896	30.917	53	0.429	0.116	2.497	34.515
64	0.179	0.057	2.960	32.143	144	0.449	0.162	1.913	20.301

Table A.73: K-Means - Node2Vec + Word2Vec + TransE